User Interface Standards

December 2004, Version 3

NCICB User-Interface Standards Group

Debra Martin (SAIC)
Jessica Rodgers (SAIC)
Clint Malone (SAIC)
Christophe Ludet (Oracle)
Shaji Kakkodi (Oracle)
Kalpesh Patel (Ekagra)
Kim Diercksen (Terrapin Systems)
Matthew Walker (Terrapin Systems)
Smita Hastak (Scenpro)
Jackie McNulty (Capital Technology Information Systems)
Vivek Pillai (Capital Technology Information Systems)
DeShera Lassiter (Booz Allen Hamilton)
Daniel Hower (Booz Allen Hamilton)

Please direct questions and/or comments to;
NCICB Application Support
301-451-4384 or
888-478-4423
http://ncicb.nci.nih.gov/support
ncicb@pop.nci.nih.gov

Preface

This document illustrates the User-Interface (UI) standards that should be followed and enforced in existing and new NCI applications and websites. It is intended for use in conjunction with the NCICB User-Interface Library which will assist in the development of NCICB web tools.

This document is intended to be a "living" document; periodic reviews of the standard will be conducted to continually enhance its usefulness, appropriateness, and practicality. More specifically, the standard will be modified as necessary to accommodate comments, concerns, and difficulties of users and developers.

Please note: These standards should be implemented as a whole. Without exceptions, all applications must comply with 508 guidelines.

1 Introduction

1.1 Purpose

There are three primary objectives of this User Interface (UI) standards manual;

1. Establish a standard look and feel for all online entities of NCICB

2. Define standard and consistent behavior for UI objects to make it easy for users to reuse their experience.

3. Review and implement 508 Compliance guidelines

These objectives will allow users to be more productive in each of the applications because they will have a consistent foundation of navigation, behavior, and appearance. Concurrently, an effective user interface standard will support simplicity and consistency across applications while satisfying user expectations for system behavior and response. It is intended as a reference manual and provides details about UI Standards, such as:

- Colors
- Screen Resolutions
- Page layout
- Headers, Footers
- Basic Website Components (Window Titles and Menus, Action Buttons, Record Controls, Navigations, and System Info)
- 508 Compliance
- Data Screen Components (Block Placement and Labels, Pop-Lists, List of Values, Radio Buttons, Check Boxes, Buttons, and Single / Multiple Record Blocks)
- Screen Navigation

This UI Standards Document is to be used by all individuals who are designing and developing web-based applications for NCI. The primary purpose behind enforcing a standard for user interface development is simply to promote visual and functional consistency across all applications comprising the NCI e-community. As more and more applications follow these guidelines, users will be able to transfer skills and experiences from one application to the next.

2 Using this Document

The remainder of this document will present the details of the UI standards, and is intended to be used in conjunction with the NCICB User-Interface Library. The use of this document differs slightly depending upon the role of the individual on the project:

Developers should use and abide by these standards when designing and developing new sites and applications. Developers should also use this document to modify existing NCI applications.

Team Leaders should use these standards during code reviews and unit testing to ensure that the standards are followed and their applications are consistent with all other project applications.

Quality Assurance Staff should be familiar with these standards so they can enforce the integrity of the standards across all screens in an application and across all other project applications. This will ensure that all aspects of the objects that comprise an application screen are specifically documented with respect to the details that developers must follow in developing consistent, standard screens.

Project Managers will periodically review applications to ensure that they conform to these standards; evaluate the effectiveness, currency, and acceptability of the standards; and make modifications as necessary.

The standards discuss all aspects of the objects that comprise an application screen and explicitly document the details that developers must follow in developing consistent, standard screens. These include the following:

- Banner and Menu Screen structure
- Basic Processing Screen structure
- Screen Objects including:
  - Window Titles
  - Push Buttons
  - Tool Bars
  - Menus
  - Lists of Values
  - Radio Buttons
  - Scroll Bars
- Screen Navigation
- Mouse Usage
- Block Labeling and Color Changes

3 508 Compliance

In 1998, the Rehabilitation Act was amended by the Federal Government to strengthen Section 508 of that Act to require that electronic and information technology procured by the Federal government be accessible to people with disabilities. The law applies to all federally funded programs and services and to all Federal agencies when they develop, procure, maintain, or use electronic and information technology. Federal agencies must ensure that this technology is accessible to employees and members of the public with disabilities, unless that agency claims that this would pose an undue burden and files a written statement of burden. Section 508 speaks to various methods of disseminating information, including computers, software, electronic office equipment, and Federal web pages. Section 508 does not apply to web pages belonging to private companies.

> *How do users with disabilities access my website or application?* People with permanent disabilities, like hearing, visual, physical or cognitive impairment, access the internet using a wide variety of alternatives. Some individuals rely on speech output; while others rely on Braille output or screen magnification, most of which will use some type of screen-reader software. In any case, our goal is to use 508 Compliance standards as a guide to make our sites and applications as accessible as possible to everyone using the internet.

The remainder of this section discusses the specific web accessibility guidelines, necessary to comply with Section 508. Theses guidelines will:

- Provide specific direction for the universal accessibility of web pages, applications, data and information for employees, business partners, and customers, regardless of their abilities or disabilities.
- Help ensure that web-enabled information is compatible with assistive technologies.
- Establish a framework that can continue to evolve as Federal guidelines are solidified and information technologies advance to meet the needs of all individuals.

3.1 Color

The use of colors is only advantageous to those users that are not visually challenged. Individuals with limited vision, no vision, or color blindness will not be able to access or distinguish information that is identified only through color. To ensure compliance with Section 508 guidelines and support access by the visually impaired, applications must be designed so that all information conveyed by color is also available without color (e.g., from context or markup).

Please note: It is required that all statements in which information is conveyed solely through color will be restated or supplemented by statements that provide the information without reference to color.

> **Example:** The statement "All fields listed below in red are required." conveys information solely through color. The statement should be reworded to state "All fields listed below that are marked with an asterisk (*) are required." These asterisks may themselves be red, but color may not be the sole identifier of required fields.

3.2 Graphics

While graphics can clearly identify features and objects available from a screen and may often provide a very intuitive illustration for these features, graphic items cannot be relied upon to solely identify these features. To conform to Section 508 guidelines and support access by the visually impaired, we must couple any graphic items with an associated text equivalent, called an ALT Tag, which provides a simple textual description of the purpose of the non-text (graphic) element. Screen readers, screen magnifiers, Braille readers, and others depend on the textual descriptions for non-text elements to ensure access by those individuals that require assistive technologies.

3.2.1 ALT Tags

When developing an application, provide a text equivalent for every non-text element. When an image indicates a navigational action such as "move to the next screen" or "go back to the top of the page," the image will be accompanied by the actual text that states the purpose of this image. This can be accomplished by the use of the ALT tag (50 character limit recommended) or the addition of linked text that points to the same location as the image into the page immediately preceding or following the graphic. More complex images or graphics can be made accessible by the use of the LONGDESC tag or the "D" link, which permit longer descriptions than the ALT tags.

"D" link

The "D" link is an interim accessibility solution established by the CPB/WGBH National Center for Accessible Media. The convention is to place a single text character "D" near (preferably, above, or immediately prior to) the image. The "D" should be the same color as the background of the page, so that those not accessing the application via a screen reader or other alternate browsing application are not aware of the link's existence. The "D" should link to a separate page containing a textual description or access of the image or functional with which the "D" link is associated.

LONGDESC tag

The *LONGDESC tag* will allow users to access a separate, detailed text description of the content. Longdesc, though part of the HTML specification, is not widely supported by browsers. The target of this link should be a separate web page which displays the long description in text-only format. Once the user has read this, they can then return to the page with the image.

Example:

For a graphic of a house:

```
<IMG SRC="home.gif" ALT="Drawing of a house.">
```

For a graphic of a house that links to the home page:

```
<A HREF="home.htm">
<IMG SRC="home.gif" ALT="Link to the home page.">
</A>
```

For a graphic of a chart that needs additional explanation via a LONGDESC tag:

```
<IMG SRC="graph1.gif" LONGDESC="graph1.htm" ALT="3-d sales chart.">
```

3.3 Skip Navigation

It is common for designers to place a consistent set of navigational links across the top, bottom, or side of every page to facilitate easy navigation through the site. This technique can render the use of the site tedious and difficult for persons using a screen reader, as a screen reader moves through the page from top to bottom, and re-reads the same list of links each time a new page is loaded. Inserting a skip navigation link allows these users to move directly to the content of the new page.

The application shall provide a method to facilitate the skipping of repetitive navigation links by those users with assistive technologies. To facilitate access to page content, insert either a text link or a graphic with a text description before a repetitive navigation link set so that users can skip directly to the content of the page. Be sure to include a target or anchor to which the link should skip.

Example:
skip over navigation hat.
<! -- navigation coding goes here -->
<a name="post-art">First text of following main body...

3.4 Flicker Rate

Some individuals with photosensitive epilepsy can have a seizure triggered by displays which flicker or flash at certain frequencies. To be fully Section 508 compliant, our applications must be designed to avoid screens that flicker with a frequency greater than 2 Hz and lower than 55 Hz. In general, avoid graphics that move, flash, continually refresh, rotate, or move text across a screen.

3.5 Frames

While Section 508 guidelines do not preclude the use of frames, they are typically avoided because great caution and considerable design attention must be expended to ensure that the frames are laid out in a manner that ensures easy use by visually impaired individuals. Please review additional information regarding frames in the Appendix.

3.6 Table Construction

Row and column headers allow screen reading software to properly resolve data tables that present quantitative or qualitative information. When the tabular display requires a check box to support the selection of an individual row, that check box field should appear as the left-most column. It will be toggled either through direct change by the individual via the mouse or keyboard, or through the use of the “Select All / None” controls.

Users of screen readers can easily get “lost” inside a table because it may be impossible for them to associate a particular cell that a screen reader is reading with the corresponding column headings and row names. To assist in the display of tabular information for the visually impaired, the application should rely primarily on text tags to present data rather than tabular presentation.

Example:
<TR>
<TH id="t1">Name</TH>

```
<TH id="t2">Cups</TH>
<TH id="t3" abbr="Type">Type of Coffee</TH>
<TH id="t4">Sugar?</TH>
</TR> <TR>
<TD headers="t1">T. Sexton</TD>
<TD headers="t2">10</TD>
<TD headers="t3">Espresso</TD>
<TD headers="t4">No</TD>
</TR>
<TR>
<TD headers="t1">J. Dinnen</TD>
<TD headers="t2">5</TD>
<TD headers="t3">Decaf</TD>
<TD headers="t4">Yes</TD>
</TR>
```

Table 1 shows a table with four columns and two rows:

Table 1 Example of a Table

Name	Cups	Type of Coffee	Sugar
T. Sexton	10	Espresso	No
J. Dinnen	5	Decaf	Yes

A screen reader would render this table in the following way:
Name: T. Sexton, Cups: 10, Type: Espresso, Sugar: No
Name: J. Dinnen, Cups: 5, Type: Decaf, Sugar: Yes.

3.7 Testing Compliance

To fully ensure Section 508 compliance within our applications, the use of automated accessibility compliance testing will be necessary as well as the use of additional individual user review. The automatic checking software will review each graphic tag to locate a completed ALT tag, and will flag graphics that do not have an associated ALT tag. Testing should verify that a text equivalent is available for every non-text element (e.g., via "alt", "longdesc", or in element content). The use of multimedia formats will also be flagged as non-compliant code.

Individual users should:

- Review all repetitive page links using either a text-only browser or a graphics-enabled browser with the graphics disabled.
- Ensure that each skip link has been inserted and that the skip takes the user directly to the content of the page.
- Examine the links apart from the body of the document and ensure that the descriptive text still makes sense when read out of the context of the document.
- Test that all forms are accessible – that labels, text descriptions, tabbing order, etc. are correct – by using a screen reader and keyboard-only mechanisms.

4 Overall Standards

This section provides guidelines on some overarching standards. It defines the supported standard configurations, browsers, colors, resolution, etc. It also defines the standard configuration and environment supported by the user community and the minimum subset that must be considered by the development team when designing and developing web-based solutions for the user community.

4.1 Internet Browsers

Applications should be designed such that no browser modifications are required to display the application content in its optimal presentation. If a plug-in is required to support the browser, links should be provided to obtain the necessary modifications. Also, pages should be designed so that they do not require horizontal scrolling when displayed at the optimal screen resolution of 1024x768 pixels.

4.1.1 Browser Requirements

Design your application so it will render in the NCICB-approved browsers. The list of approved browsers will be reviewed periodically as new browser versions are released. The current list of supported browsers includes:

- Microsoft's Internet Explorer, Versions 5.0 and above
- Mozilla 1.0

Please note: It is important that all developers be aware of their user group. If requirements exist for additional compliance (example: Safari 1 and up), test for those users and ensure the same level of functionality as the supported browsers.

4.2 Color Schemes

The judicious use of colors, shading, and highlighting are valuable mechanisms to assist in visually orienting the user to the layout of the screen and context of the data. Colors can also be used to visually group like items and radically highlight errors; however, all statements in which information is conveyed solely through color must be restated or supplemented by statements that provide the information without reference to color. [Please refer to Section 2. 508 Compliance, and more specifically to the subsection that addresses color and 508.]

The NCICB User-Interface Library contains recommended color schemes, which have been designed to compliment the NCICB interface, as well as that of Cancer.gov. Please select and implement only one color scheme.

Please note: The diligent and systematic use of cascading style sheets throughout the application can simplify the consistent use of color and fonts. Standard style sheets are also available in the NCICB User-Interface Library. They are included in the template directories.

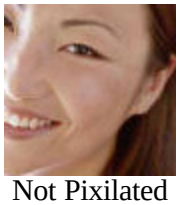
4.3 Screen Resolution

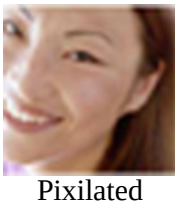
The screen layout should be flexible for different screen sizes from 800x600 pixels and up. All screens should be optimized for 1024x768 pixels resolution.

4.4 Graphic Requirements

In an effort to create consistent, visually appealing graphics for all of NCI's websites and applications, avoid using pixilated, improperly compressed, or improperly scaled graphics. Figure 1 is an example of two pictures, one which appears to be pixilated and one which is not distorted.

Figure 1 Pixilated vs. Properly Formatted


Not Pixilated


Pixilated

To further ensure the use of visually appealing graphics, set all graphic dimensions in the graphic's HTML tag.

Example: <img src="nscclogo.gif" height=50 width=50>

This tag will reinforce the dimensions of the graphic and prevent further distortion which may be linked to site construction.

There are several compression formats used and recognized by web browsers; however the two most used and widely accepted compression formats are gifs and jpegs. Choosing the appropriate compression format will also ensure visual integrity. The following table has been created to give those without an existing understanding of graphic compression a quick overview. Table 2 should give you the information you need to choose the appropriate compression format.

Please note: The standard resolution of any web graphic is 72dpi.

Table 2 Choosing the Appropriate Compression Format

	JPEGS	GIFS
Used For	compression for photographs and images with many color fields	drawings and images that have large blocks of color
Color Restrictions	jpegs use millions of colors	restricted to 256 colors so they are not best for intricately colored images
Advantages	complex color arrangements where colors blend into each other are best compressed as jpegs	can be transparent and animated

Always provide a text equivalent for every non-text element by method of an ALT tag. You can see more about ALT tags in the 508 Compliance section of this document.

4.5 Style Sheets

A style sheet is made up of various style rules that tell a browser how documents are presented on screens and in print. By attaching style sheets to structured documents on the web (e.g. HTML),

authors and readers can influence the presentation of documents without sacrificing device-independence or adding new HTML tags. Style sheets will help create a consistent look and feel on all of NCI's websites and applications. Please refer to the Appendix to review additional recommendations on how to implement style sheets.

4.6 Use of Frames

While frames provide an easy way to organize items on a screen, they can create problems for users because different browsers use and display frames in different ways. Because of problems defining and predicting behavior and interdependence within the frame objects and the standard browser functionality (back and forward button), it is advised that developers *avoid the use of frames* unless justified by specific user needs.

4.7 Use of JavaScript

The validation of fields will be performed with a server side object to ensure that data integrity and validity is not compromised before data is inserted into the database. JavaScript cannot be used in place of business validation which must occur on the server side.

At a minimum, each screen should be conducting field-level validation through a server side object. The decision to use JavaScript to conduct field-level validation in a timely manner, at the time values are entered and fields are navigated, will remain with the development teams and the joint review committees and be supported only when there is a strong driving requirement for such validation arising from the user community.

4.8 Columns

When designing to maximize screen real estate by placing data side-by-side, the data should be presented in a columnar manner with rows and columns aligned. When feasible, ensure that user input controls (e.g., text boxes, drop-down lists, etc.) are of an identical length when presented in a columnar format. This will allow for a more pleasing visual experience for the user.

4.9 Language Support

4.9.1 Standard Language

The basic language for all the NCICB websites and applications is United States English.

4.9.2 Multi-language Support

There is currently no requirement to support any language other than United States English.

5 Page Layout

This section defines the standard structure, components, and behavior for all NCI pages. All the pages are divided into several areas. This section describes the overall layout of each page, the location of each area and the position of the components within each area.

To facilitate the use of our applications, the basic screen structure is designed to provide for a flatter navigation style which allows users to easily navigate to features that are directly associated with the current screen. This increases the intuitiveness of the system and eliminates the need for complex pull-down menus. Specifically, screen layouts will meet the following goals:

- Provide a main navigation.
- Provide application-specific controls.
- Provide intuitive navigation with minimal training.

To achieve consistency, all teams must follow the same design principles for menu layout, action button layouts, window design, and form behavior. The flat navigation will make the navigable path shorter and the overall behavior or the system more consistent and predictable to end-users.

Please note: This section is intended to provide guidelines for standard user interface, exceptions will be made for transactional applications.

5.1 Basic Layout

Every web-based screen will have the same underlying structure. While some of the content will change depending upon the purpose of the screen the basic layout should be consistent. A page will have a window title and five areas. These areas include:

- Header Area
- Sub-navigation
- Main Navigation
- Page Content Area/Action Area
- Footer Block

The following graphic illustrates the overall layout and the location of each area.

Figure 2 Basic Layout

5.1.1 Header Area

The header for every page will contain the Cancer.gov mini banner, as specified by cancer.gov web resources Design Standards, *http://webresources.cancer.gov.* The Cancer.gov mini banner should be on every page of your site and must link to the Cancer.gov home page, *http://cancer.gov*. The header must also contain the project logo and the NCI logo. The project logo should align left and the NCI logo should align right.

5.1.2 Navigation

The menu area separates the Header from the page content area. It should be composed of a combination of navigation items that operate as a single action that navigates directly to another base screen, application, document, or website.

Properties:

- Each navigation item will have a distinctive and unique text word or short phrase identifying its category or action.
- The text will be centered within the navigation item area.
- The Help item should always appear as the right-most item.

There are certain default items that should appear in every menu. There should be an average of nine primary navigation buttons (+/- 2). The sub-navigation will be application-specific and

placed to the left of the page content area/action area. If your navigation exceeds three or more levels, the use of breadcrumbs is recommended. For more information on breadcrumbs, please consult the appendix.

In addition, menu items will be displayed based on access privilege. For example, the external home page menu bar does not contain navigation access to any internal applications or options that require a username and password. It is not until a user completes a successful connection to the database that the complete menu bar is displayed with the internal application items.

The default menu items are illustrated in Table 3.

Table 3 Default Menu Items

Menu Item	Submenu Item	Description
Home		A navigation button back to the application home Page
Help	Documentation	Allows viewing and printing of on-line manuals
	About Box	Gives the application's name and version number
	Other Possible Help Items:	
	Help	Context-sensitive help
	Display Error	Displays most recent error message, if any

If possible, the submenu should have a visual change to reinforce which menu choice is being selected. Our standard approach will be to make the submenu items a different shade of the same color as the main menu and simply change the color of the selected main menu item to visually identify which main menu item is associated with the submenu.

5.1.3 Page Content Area

- This area will be customized on the individual screens.
- This area is used to present most of the data processing content. Technical leads can determine the layout of the page content area within the constraints of the style sheet, but keep the following in mind.
- Data fields are typically laid out in a free-form manner that promotes easy identification of items through grouping, aligning, and highlighting.
- The order of fields should be familiar to users and support a comfortable navigation between those fields.

The bulk of the display portion of a screen is comprised of data blocks that support the primary function of the screen. Fields usually derive from a single database record but can include multiple records or data from several tables. The blocks that are included in the page content area are of two basic types: multiple row blocks which display many rows of data in a tabular format, and single row blocks which position individual data fields in logical fashion across the display screen. For more information on data blocks and the fields that are used within them, see the Section 7 Screen Objects.

If the errors are generated due to the server side validation, a user-friendly error messages will be displayed on the top of the page content area with the original content that needs modification.

5.1.4 Action Area

Between the page content area and the footer area resides the action area, which contains any action buttons applicable to this screen. Action buttons are push buttons that allow users to perform specific operations and system functions with respect to the data on the screen.

Properties:

- Each action button shall be represented consistently with respect to size, shape, color, fonts, etc., except where there is a need to have the user visually focus on a specific action button.
- All buttons will have same height and unified width to accommodate various label text lengths.
- All action buttons will contain a label.
- Action buttons that can cause the user to lose significant amounts of data, such as a 'Delete', should have a confirmation message associated with it, to ensure that users actually intended to perform the destructive action.
- All the action buttons should use the standard button color. All buttons should be displayed with the default background with a black font.

5.1.5 Footer Area

The footer layout should be consistent with cancer.gov design standards referenced above. Similar to the header area, each page will end with a standard footer area. Items for the footer area should include standard items such as:

NIH Links of Interest:

- Contact email for user feedback or problems
- Cancer.gov, NIH, HHS, FirstGov logos with hyperlinks
- FAQ
- Contact Us
- Privacy Notice
- Disclaimer
- Accessibility
- Application Support

Figure 3 is a representation of how the footer looks when implemented.

Figure 3 Footer

5.1.6 Disclaimers

The home page should include a disclaimer addressing privacy and retention of information collected.

> **External Links** -- web pages containing links to external web pages should include a statement that releases NIH from responsibility for the material included. It is important to avoid giving the impression that NIH is endorsing external information or products.

Privacy Policy – Disclaimers should be included to advise the public of intended use of the information learned about site visitors.

5.1.7 Screen Title

The title bar simply lists the application/website name and NCI. Optionally, it also displays additional identifying information about the screen (e.g., an indication of the screen's function).

5.2 Default Information

Many of the applications are based on a business that revolves around a fairly constant set of default information for a user. The default information area provides a consistent, locatable display area for this default information that serves as a reminder to the user. There are several different kinds of default data that can be displayed in this area. Some of these include the following:

5.2.1 Log On

User Information: The right half of the header area shall contain the user's name, institute code/institution name (if needed), and user's role (if needed) in a bold font. In addition, many of the business area applications are based on a business that revolves around a fairly constant set of default information for a user. This information can be as simple as their user identification information, but could be something more complex as default meeting identifiers. The default information area provides a consistent, locatable display area for this default information that serves as a reminder to the user.

5.2.2 Log Off

This default information is usually simply a "Log off" link that is displayed in the title block just below the user's information. This provides a consistent location for users to locate the exit link. Prior to a user's successful connection to the network, this area shall display some standard name, such as "Guest."

All applications should include a lost password link to NCICB Application Support, ncicb@pop.nci.nih.gov. *Please note: If the application does not require log-in, this standard is not applicable.*

The layout of the page is designed to be flexible, giving your team the freedom to implement the logon screen to the right side of the design or more centered in the action area. Figure 4 is an example of how the application logon field can appear.

Figure 4 Log on

5.3 Alternative Pages - Text-Only Template

Our application screens will provide text-only alternative pages to provide equivalent information or functionality to visually impaired users in cases where no other suitable alternative exists to provide accessible functionality. The content of the text-only page shall be updated whenever the primary page changes.

As a last resort, if other options to provide accessibility to content do not exist, designers should construct a text-only page to provide alternate access.

Please note: The use of a text-only alternate page is discouraged. They should be developed ONLY when there is no other alternative.

In creating a parallel text-only page, the designer is committing the maintenance staff to double the time required to maintain this page, as both the primary and text-only alternative page will require updating each time a change is made. Every effort should be made to find an accessible alternative before the construction of the text-only alternate (i.e., design the primary page to be fully compliant to eliminate the need for an alternative page). Any web page for which a text-only alternate is provided should be reviewed to ensure that no alternate accessible option exists, and that the text-only implementation follows guidelines by providing full access to the functionality on the non-accessible page.

6 Application Home Design Standards

The construction of the application home page will consist of the basic layout as described in Section 5, but will also include the option of an application/website banner that can further brand the project initiative.

Below is an example of how the application home page can be constructed. This option includes the application logon field along the right side of the screen above the "What's New" and "Tip(s) of the Day" sections. The application logon field can also be placed in the center of the action area, especially if all content must be secure and read only after a successful connection has been completed.

Figure 5 Application Home Page

In addition to the application banner, the application home page should also include links to:

- About this Application (required)
- What's New (optional); a section for the latest news affecting the user community
- Tip of the day (optional)

6.1 About this Application

This feature is required. About this Application should explain what the application does and how it is used is. This section should also include a version number. This is not a help section, and it should not be treated as such.

6.2 What's New

This section appears on the right half of the page content area. It contains a display of information for the user community about new and exciting activities going on in and around this application and their community. It also provides a means for the project team to communicate to the user community changes in policy that may affect their job and/or their use of this application.

- The contents of this section will derive from the database to permit easy addition, update, and modification of the contents. These may be entered and controlled by a specified group of users who have appropriate permissions.
- The text that displays will be limited to one hundred (100) characters to give the user a flavor of the item.
- A hyperlink will be associated with each item to provide users who are interested with a complete listing of the item.

6.3 "Tip(s) of the Day"

To promote the continued education of the users on the use of the available tools NCICB can offer a "Tip(s) of the Day," which should provide a helpful recommendation that simplifies the use of the application. The contents of this section can be derived from a database to allow for easy addition, update, and modification. These may be entered and controlled by a specified group of users who have appropriate permissions. Please pay close attention to the following guidelines when creating your application's "Tip(s) of the Day."

- The text that displays will be limited to one hundred (100) characters to give the user a flavor of the item.
- A hyperlink will be associated with each item to provide users who are interested with a complete listing of the item.
- The "Tip(s) of the Day" should be called randomly so that users are presented with a different tip with each log-in.

6.4 Screen Id

The screen id is a string that uniquely identifies the screen the user is currently on. This identifier is very important to both development and quality assurance staff in reporting, investigating, testing, and approving changes, problems, and difficulties. So that the identifier is easily found by the user it should be located in the bottom left-most corner of the content area.

6.4.1 Formatting the Screen Id

The format for the screen id is: XX9999 where XX is the two or three characters, which have been approved as an acronym that represents the application (e.g., CM, REV, RR, ICO, IPF, etc.) and 9999 is the number associated with the object as it is stored in the configuration management library. This should provide each page with a unique id, which, in turn, will supply the user with a method of determining their location in the application. This information is useful when seeking

help from application support and when identifying errors. *Please note: The number of characters which represent the application (XX) can exceed two or three characters; however the id format should be consistent across of all the application.*

7 Screen Objects

Some NCICB applications serve to support data query, entry, update, and/or reporting needs of our user community. As such, these screens are primarily composed of two types of fields: database fields that arise or are derived from the database and displayed on the screen, and text labels that identify the associated database field. There are also additional screen objects that support the screen operations (e.g., icons, buttons, graphic items, etc.). This section provides the guidelines for use of these fields.

7.1 Data Fields

The data fields typically arise from the database and may constitute either a direct display of a database field or a derived representation of data. To ensure a consistent presentation of database information, database fields must conform to the following guidelines:

- Database fields should be associated with a text label unless the field is completely intuitive and easily comprehended without the use of a text label.
- When database fields are presented in a tabular or columnar format, each field should be clearly delineated from other fields in the row and column. There are two acceptable ways to highlight the field display: either outline each field with a display box or alternate the shading of rows to distinguish them from each other.

Most of the fields are derived from the database and fit into one of the following categories.

Table 5 Field Categories

Field Type	Description	Acceptable Formats
Date	Include the date format in the field label.	mm/dd/yyyy (e.g., 1/1/2001)
Date Time	Similarly, include date, time and time zone format in the field label.	mm/dd/yyyy hh:mi TIMEZONE (e.g., 1/1/2001 14:30 EST)
Telephone Number	Include the phone format in the label. It is recommended that the U.S. phone format is used, however, please allow for multiple formats if necessary.	xxx-xxx-xxxx
Name	The name is comprised of the First, Middle and Last name fields. It is recommended that the First Name, Middle, and Last name be separate fields and in this order. The relationship may change and will depend on how the information is used.	First Name, Middle, and Last name

General rules for data within in the fields include the following:

- Text fields should be left justified.
- Fields containing numeric or decimal data should be right justified. However, dates can be optionally positioned under the calendar icon.
- Fields containing code values (numeric or character) can be left justified if display is improved. Outline data fields with a plain box or bevel down box.

- Query Fields: used for searching, have a white background with black text. Display or Read-Only Data. All Display and Read Only fields should be labeled with a gray background to distinguish them from other fields.
- Editable Fields. If the field value can be entered or modified by a user, that field should be displayed with a white background and black foreground (i.e., black font text). Field labels should be marked with a '*' for mandatory fields.
  It is recommended that when applicable, query fields should contain links for the first and last pages of the data table.

7.2 Data Blocks

The bulk of the display portion of a screen is comprised of data blocks that support the primary function of the screen. Blocks are of two basic types: multiple row blocks which display many rows of data in a tabular format, and single row blocks which position individual data fields in logical fashion across the display screen. The remainder of this section addresses the common elements on data screens and features specific to the multiple and single row data blocks.

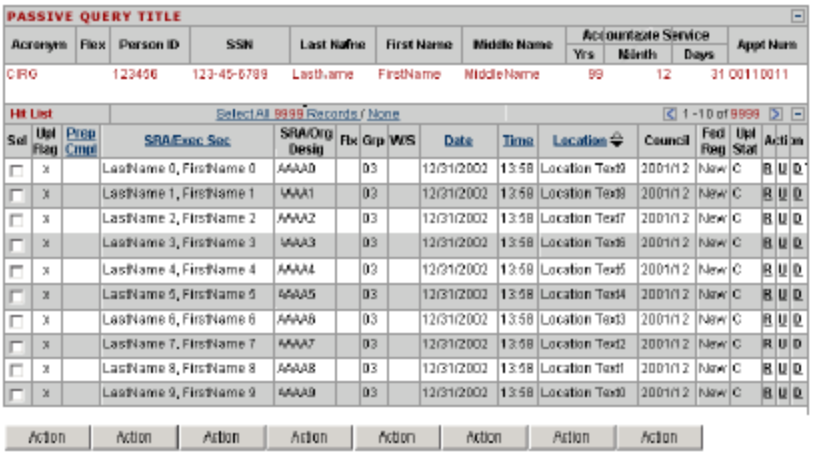


Figure 6 Data Block

7.2.1 Block Separator

Each block of data should be preceded with a horizontal title bar that separates this block from any previous blocks. This block separator contains information used to highlight the block of data below and support easier navigation between pages of data in the accompanying hit list. Specifically, the block separator line is comprised of the following:

1. **Block Label**
   Each visual block should be uniquely identified with a block label in the upper left corner of the visual block. The text for the block label should appear in a single horizontal line that spans the width of the visual block. This provides for easier mental separation of data groupings. The font, color, and display properties of the block label are recorded in Section 8.

2. **"Select All / None" Control**

Some of the hit lists are designed to include a check box for each row that is used to support various functions to be applied to selected rows. For instance, a check box may be used to identify rows for which a user wishes to print a report. Because hit lists can get large, the “Select All / None” controls appear on the block separator to the right of the block title to allow users to quickly select or unselect all of the hit list entries. These controls are simply hyperlinks which, when clicked, will select or unselect the main check box for each row. *Please note: "Select all" could have undesired user results.*

The third option listed below is an optional feature for future applications.

3. **“Select All in a Page” Control**
   In a scenario where a database query returns 1000 results, but the results are only displayed 10 per page. The "select all on page" button would check the box for the 10 results on the current page, while the "select all" button would check the box on the 10 displayed results and also the 990 results not appearing on the page.

The focus of these controls is the entire hit list. The “Select All” hyperlink will cause all the check boxes in ALL hit list records to become checked or selected. The “Select All in a Page” hyperlink will cause all the check boxes in ALL hit list records to become checked or selected, even if they are not displayed on the current page. The “Select None” hyperlink will simply uncheck all rows that have been checked or selected.

If the screen functionality does not call for selecting or unselecting more than one record at a time, this control object should not be displayed in the block separator. Radio buttons can be used instead of check boxes for exclusive selection.

Record Navigation Controls

Please note: This control is optional and can be hidden from view if it is not necessary. Hit lists that contain more data rows than can appear on a single display screen are divided into discrete pages of information, rather than presenting the user with one long page of results. To support the presentation of separate pages to the user, include the paging controls and location labels in the block separator.

The right-most side of the block separator will contain the record location information, the location of the displayed records relative to the full hit list (e.g., 1-10 of total hits x). To the right of this location information will be two buttons that support the navigation to the preceding and following page of hit list records, respectively. For example, this control must be used when the data display spans multiple pages. Its use is only necessary when the underlying block is a hit list that contains more rows than are displayed on the screen at the time. When the hit list displays the first page, the left arrow button should be hidden. When the hit list displays the last page, the right arrow button should be hidden. In the event of a large data set, first page and last page buttons should be included in the navigation controls.

Collapse Control

The collapse control will allow the current block to be condensed into a single line represented by the block separator. The button in the right corner will toggle between the standard windows expand and collapse buttons depending upon the state of the current block. This feature will allow hit list blocks to expand in the relative screen space and display additional rows of valuable information. This feature is optional and can be used at the discretion of the developer.

Figure 7 Collapse Control

7.2.2 Block Placement

The layout of the contents of a data block must conform to the page layout standards set forth in this document. The actual placement of blocks with respect to each other and the display page content area is subject to consensus and approval by the user community, conformance with these user interface standards, and adherence to Section 508 compliance considerations.

7.2.3 Empty Data Block

If a search returns no results, the table headers and one empty row will still be displayed. The empty row will not contain a check box or any action links.

7.3 Field Labels

Data items are identified with a label. General rules for field labels are defined in the following Properties sections.

7.3.1 Auto Complete

The auto complete attribute for all text boxes with sensitive information (i.e., user credentials, patient identifiers) should be set to "off."

Example: <input type="text" autocomplete="off">

7.3.2 "for" Attribute

All labels tags should include a "for" attribute, which ties the label tag to another label or field.

Example: <label id="lblName" for="txtName">Name</label><input type= "text" id="txtName">

7.3.3 Justification

Field Labels should appear to the left of a database field when the screen is constructed in a single record format (i.e., fields are positioned at various places on the canvas instead of in a tabular format). In general, the field labels should be left justified with each other to promote readability. When the screen presents data in a tabular or columnar format, text labels should appear at the top of the column, centered on the data. No colons should be used.

7.3.4 Mandatory Field Labels

Fields that must be filled in (by user or system) are identified with a bold field label AND an asterisk (*). The bold option will be a visual indicator to those users that do not have visual impairments. The asterisk is to be used to ensure Section 508 Compliance so that visually impaired users that utilize screen readers can also identify the mandatory fields. Additionally, an explicit descriptive message should appear on the page, defining the bold text and asterisks as indicators of a required field. The phrase "*Required Field" is the recommended message.

7.3.5 Optional Field Labels

Data fields that do not require a value are identified with a regular font. *Please note: Data fields that are not required (mandatory), but become required based on other data fields.*

7.3.6 Descriptive Text for Controls

Descriptive text, such as examples of specific formats to be entered, should appear to the right of the control. The location of the descriptive text should be consistent throughout the application.

7.3.7 Sort Labels

Sort labels appear (in a tabular display) at the top of data column and provide a means of sorting the data in the specified column. Hyperlinks are used to indicate that the column can be used to sort the displayed table. When a column is selected as the primary sort, the table will be resorted according to the data in that column, the sort label should change to include an upward or downward arrow to the right of the column label indicating that this column is being used to sort this table. Use an upward arrow for ascending sorts and a downward arrow for descending sorts.

By default, the first use of a sort label will sort the data in the hit list in ascending order according to the criteria defined for that sort label. This is typically just the contents of the selected column but may also be a combination of fields that appear on the screen. If the header of a primary sort column is clicked again, the table will be resorted in opposite order.

Only one sort can be applied at any particular time. The column that is currently controlling the sort of the hit list will be the only one identified by the upward or downward arrow.

7.3.8 Mouse Hover

Upon mouse hover of an action or navigation item (i.e., button, image, etc.), the cursor image should alternate to a hand to indicate that the item is clickable.

Example: <input type="button" style="cursor:hand">

7.4 Image Maps

Image maps specify the coordinates within a graphical image ("hot spots") within which a specific link may be accessed. They are often used as a creative visual means for presenting site maps and related links to the users.

Server-side Image Maps

Screen readers cannot read images generated by server-side scripting, which specify the coordinates within a graphical image ("hot spots") within which a specific link may be accessed.

Client-side Image Maps

Client-side scripts, however, resolve on the browser end, so there are more options to provide accessible alternatives if you choose to use a client-side image map. Provide redundant text links for each active region to support the visually impaired.

Requirement: Do not use server-side image maps.

7.5 Icons

When implementing icons, follow these guidelines:

- Icons should be large enough to be viewed on a screen with the standard screen resolution and should be labeled intuitively.
- As a general rule, icons should be avoided unless they clearly and distinctly illustrate a function or capability that does not require a text description.
- Icons that are repeated across pages should be positioned in the same place on each page on which the icon is displayed.

7.6 Drop-Down Lists

Drop-down lists (also referred to as pop-lists) are used to allow users to select one value from a known list of valid choices. When activated by clicking on the list down arrow, a display window is presented just below the current field (like an extension of the field with possible values). The current field value is highlighted and users simply move the cursor to the desired value. If selection can be multiple values, use a list box.

Once expanded, drop-down lists should strive to let users see all of their choices in one glance. The drop-down list should be limited to no more than 15 items. The contents of the list should be ordered according to some logical convention, and in the absence of a specific convention, the list should be alphabetized.

Drop-down lists are used when a field has a small, very static list of possible values. Since these kinds of lists are resolved when the screen is rendered, they become part of the screen code displayed, so the size of the lists must be closely monitored and controlled. If more than fifteen (15) items are needed or the list requires periodic modification, the valid values should be stored in a look-up table and displayed using a List of Values.

7.7 List Boxes

List boxes are used to allow users to select more than one value from a known list of values. When space is available, list boxes are the preferred method of presenting a list of possible selections, as this allows the user to view several, if not all, of the possible choices without having to take a discrete action. List boxes should strive to let users see all of their choices in one glance. A minimum of three choices should be visible when using a list box. If space considerations won't allow the display of three choices, then a drop-down list control should be used instead. The contents of the list box should be ordered according to some logical convention. Common ordering schemes include:

- Frequency of use
- Importance / criticality
- Logical groupings (e.g., common elements)
- Alphabetically (to be used in the absence of other organization conventions)

If multiple-selection is available for the list box, it is often advantageous to inform users of this functionality. The standard multiple-selection techniques are often unfamiliar to the novice or infrequent users. As such, this type of information is useful as a pop-up help message.

Figure 6 Drop-Down List

7.8 Radio Buttons and Check Boxes

7.8.1 Radio Buttons

Radio buttons are typically used to allow users to select one option from a known set of mutually exclusive options. The label text should be aligned left, however the text should be placed on the rights side of the buttons.

A user chooses one of several values by clicking on the appropriate box. This technique for data entry can be used when a field can have only a limited, fixed-set of mutually exclusive values. The value stored in the database is the code assigned to the radio button column. The user does not need to know that code.

7.8.2 Check Boxes

Check boxes should be used exclusively to allow users to select one or more options from a known set of options. If the number of options becomes large, consider using a multi-selection list as an alternative control. The check box is often used to record Yes/No types of answers. Using the mouse, the user places the pointer on a box and clicks to select the item. Additionally, the space bar can be used to toggle or "flip flop" the value in the check box.

Below are examples of standard radio buttons and check boxes that should be implemented. They are both available in the NCICB User-Interface Library, with in the template directories.

Figure 7 Radio Buttons

Figure 8 Check Boxes

7.8.3 Overall Considerations

For readability, radio buttons and check boxes should be aligned (vertically or horizontally) and put into groupings where appropriate (see examples above). If needed for organization purposes, consider grouping either set of controls with a border to reinforce that choices belong to a particular group of choices. Care should be taken when choosing which of these two controls to use for a specific task, as a misuse of control types will impact user productivity and diminish transfer of learning effects across applications.

For accessibility, be sure to arrange the selection area and the label so that keyboard access occurs in a sensible fashion.

7.9 Push Buttons

Push buttons on screens are limited to the action area. Exception to this rule can be made on a case-by-case basis. This standard practice is enforced because of considerable user difficulty and confusion with having to search for buttons on screens.

These buttons must adhere to the following guidelines:

- They must activate a specific function that cannot be incorporated into an Action Button.
- They cannot navigate to a pop-up screen or related base screen. That should be handled through a navigation item.

7.10 Scroll Bars

The scroll bar is used when the list of records exceeds the number of rows available for display. The browser controls the display of the scroll bar so no further programming support is necessary for its use. The position, format, and use depend upon the browser, the number of rows or columns displayed, and the color scheme in affect. The developer should be mindful of horizontal and vertical scroll bar use and display when conducting unit testing under the supported browsers.

In general, the user communities have requested that data be controlled and displayed on a page-by-page basis rather than as a list that exceeds the length and width of the display area. This minimizes the need for the horizontal and vertical scroll bars. The vertical display area should be controlled to display only that number of records that can be placed inside the display area. The width will be more difficult to control because users will be given the option of customizing their hit lists by selecting the columns they want to appear on that list. The use of the horizontal scroll bar will probably be used more frequently as a result.

7.11 Pop-Up Windows

The following are guidelines for developing pop-up windows:

- Window Size: there is no fixed size for the pop-up windows, as long as its height and width do not exceed the base window dimensions.
- Pop-up windows are self-contained. The navigation function has to be completed within the scope of pop-up windows.
- A pop-up window should not be closeable or zoom able. To leave a pop-up window, the user has to press the "Close" or "Cancel" button.

Users cannot leave a pop-up window unless the database transaction is completed or canceled. *It is recommended that pop-up windows be avoided.* While the framework supports the use of pop-up windows, programming practices within the industry have tended to avoid their use because of potential data integrity problems that may arise. When a pop-up is launched, the server has no way of distinguishing data that arises from the original window and that which arises from the pop-up window, as requests coming from both windows will appear identical. For a flow-sensitive application, this creates problems. You now have the possibility of two sets of flow coming from what appears to be the same location.

There is, however, no part of the framework preventing a developer from using pop-up windows, but data integrity can be compromised as a result.

To be precise, though, the issue is not limited to pop-up windows in general, but specifically with opening a second or 'child' browser window. Pop-ups that are JavaScript dialog boxes or DHTML pop-ups are not an issue.

Please note: Same browser or separate window pop-ups can be used at the discretion of developer.

8 Navigation

8.1 Screen Navigation

Navigation between screen items is done with the keyboard or mouse. An item refers to an enterable field, button, radio group, check box, etc. Standard field navigation is from top left to bottom right within a block, moving down line by line. Navigation from block to block should normally proceed from top left to bottom right. However, there are exceptions made because of the use of optional fields, support for tabular numeric entries, and other constructs.

Always consider Section 508 Compliance and how screen readers will interpret the page when selecting the tab order for the fields. Other rules include:

- When navigating single record blocks, the cursor will proceed to the first item of the next block when a Next Item command is issued and the cursor is in the last field of the block or the last item of the previous block when a Previous Item command is issued and the cursor is in the first field of the block.
- If a set of buttons is grouped in a separate visual block on the right side or bottom of the screen, that block is considered the last block on the screen.
- Movement between pages should occur only as a result of a specific user action.
- The cursor should not skip display-only fields.
- The cursor should be able to be positioned on buttons.
- Auto-skip is rarely used. It will only be used in cases where the user's explicit request is based on a logical business need and an intuitive navigation pattern.
- To assist in keyboard navigation, the number one (1) should be specified as the access key for the first editable field of the form.

  Example: <input type="text" accesskey="1">

- When a new page is displayed, the cursor should be positioned in the 1st editable field. If no editable fields exist, then the focus should be given to the 1st selectable option.
- When the user presses the Tab key, the cursor will move to the next input field or action button. A page's tab sequence should be carefully laid out to ensure consistent and predictable navigation from field to field. A user should not be "surprised" at the resulting cursor navigation when the tab button is pressed.
- If possible, associate the Keyboard return with the default action button (e.g., submit) on a page.

Table 4 summarizes the cursor movements with respect to the cursor's screen position.

Table 4 Cursor Movements

Block Type	Position	Function	Movement (to enterable fields/buttons only)
Single Record	Item not in first or last position in the block	Next Item	First item to the right on the same line if one exists or first item on the line below the current position in same block
		Previous Item	First item to the left on the same line if one exists or last item on the line above the current position in same block
	First item in block	Next Item	First item to the right on the same line if one exists or first item on the line below the current position in same block
		Previous Item	Last item in the last record in previous block
	Last item in block	Next Item	First item in the first record in next block or first block when the current position is in the last block on the screen (cycle in screen)
		Previous Item	First item to the left on the same line if one exists or last item on the line above the current position in same block
Multi-record	Item not in first or last pos. in first or last rec.	Next Item	First item to the right on the same record
		Previous Item	First item to the left on same record
	First item, not first record	Next Item	First item to the right on the same record
		Previous Item	Last item in the previous record
	Last item, not last record	Next Item	First item in the next record
		Previous Item	First item to the left on same record
	First item in first record	Previous Item	Last item in the last record in previous block
	Last item in last record	Next Item	First button in the action area

9 Page Flows

A page flow is a combination of page templates whose content form some logical, common work flow. A page flow can be thought of as a recipe of page templates with customizable aspects based upon an application's detailed functionality needs. A page template is simply a pattern of instructions and guidelines that define the order and process of pages that can be reused in different applications. Page flows are typically associated with a specific type of action (e.g., uploading a file).

Examples of page flows include search/results create/add/delete/update "x"/view/update/view find/browse etc. *The remainder of this section provides visual details and examples of page templates and their specific flows.*

9.1 Page Flows

9.1.1 Message Page

Message pages and flows are used when an error, information, warning, or confirmation message needs to be displayed to the user. There are two kinds of message flows:

- An inline message - appears on the page along side the content that needs correction.
- A Message Page - the message appears on its own page, allowing the user to choose between a set of action/navigation buttons to navigate to the appropriate place in the application.

9.1.2 Search -> Hit List Page -> CRUD

This page flow defines a general pattern that allows a user to specify search criteria and generate a hit list. The user enters a search criteria in the query block indicated and clicks the Search button. The list of records, or hit list, matching the search criteria is displayed below the query block and enables the user to view detailed database rows that represent the outcome of the specified search criteria.

If the number of records matching the criteria is more than 20, then the user can navigate between the records using provided navigation. The user can then decide to Read, Update or Delete an existing record or the user can decide to add a completely new record. The changes made to any record can be saved by using the Save button.

Please note: Actions (e.g., read, edit, and delete) can be combined on one screen to improve usability pending application requirements.

Table 5 Standard Actions and Navigation

Button	Standard Action	Standard Navigation
Read	Opens the selected record on Screen 2 in read only mode	Navigates to Screen 2
Edit	Opens the selected record on Screen 2 in edit mode	Navigates to Screen 2
Delete	Deletes the selected record from the database after confirmation	Stays on same screen
Previous	Data submitted is validated and is stored on the server	Stays on same screen
Next	Data submitted is validated and is stored on the server	Stays on same screen
Add		Navigates to empty Screen 2
Search	Queries the database for the search criteria entered in the query block	Remains on the current page and displays the hit list
Cancel	Discards any pending changes and navigates off the current page to the Initiate Screen.	Navigate back to the page that initiated the action (Initiate Screen)
Save	Validates the data entered/changed on Screen N, if the data is correctly validated, the entire data saved during the course of Screen 1 to Screen N is committed to the database	If validation fails the user is returned to Screen N with the errors shown in the Error Block. Else the user is taken to the Initiate Screen

9.2 Queries and Hit Lists

9.2.1 Query Block

Located in the top most portion of the page content area, the query block supports searches for data, filtering of results displaying in the hit list, displaying additional information about the currently highlighted hit list record, etc. It may be broken into two or more blocks depending upon the complexity of the screen. The most common construct for the query block is query parameters. They provide search fields and search options that are used to tailor the retrieving and displaying of the data in the accompanying hit list.

9.2.2 Hit List Block

Located in the bottom portion of the page content area, the hit list is used to display details about the information selected or identified in the query block.

When possible, limit each hit list row to a single displayed record. This will permit more rows to be displayed in the page content area providing the users with a larger range of results to observe at any given time. Should a single display record not be possible because of the volume of data the user community requested be added to a hit list row, then multi-row displays are permitted.

Please note: Since the number of fields in the hit list will be customizable by users, the width will not be easily controlled. In these cases, the use of the horizontal scroll bar will be supported and necessary to display very wide lists.

9.2.3 Paging Considerations

Hit lists that contain more data rows than can appear on a single display screen, without requiring vertical scrolling, should be divided into discrete pages of information, rather then presenting the user with one long page of results.

Guidelines for page considerations include:

- If more than three screens of data (e.g., 50+ list items) are available, consider breaking up the presentation into separate pages of information.
- If separate pages are presented to the user, use the paging controls and location labels that are described on page 7-29 under record navigation.
- If the data requires extensive retrieval time (e.g., 10+ seconds), consider breaking up the presentation to allow for a quicker display of the initial page.
- In the event of a large data set, first page and last page buttons should be included in the navigation controls.

The shared hit list object supports customization (e.g., user personalization settings), which allow users to customize the number of items displayed in each list. This approach will allow users to make their own trade-offs in terms of scrolling, performance, etc.

Paging should be consistent and predictable across the application. Do not arbitrarily have one list break pages after 50 items, while another list breaks pages after 100 items. Controls and conventions for list management should be consistent across the application.

If separate pages are presented to the user, use paging controls and location labels in the block separator, as noted in the preceding section.

9.2.4 Search Screen Template

Filtering lists which search for specific data points can be presented in various ways. Two typical ways include a predefined filter, such as a tertiary menu, or by use of a specific search control.

9.2.5 Wizard Step-by-Step Page

Typically a wizard style step-by-step page flow initiates from another flow/object where a series of steps are required on the object. It may initiate some other flows (such as a message flow) as part of its steps. Once the steps are successfully completed, the user is returned back to the original entry point.

Screens 1 to Screen N are simple data entry screens. This kind of flow is needed if a user is required to enter more than one logical set of data. In this scenario, either all data is saved or all the data is lost, but this behavior is open to change in case of a specific requirement.

1. The user goes to Screen 1 by clicking a button or a link on the Initiate Screen.
2. The user enters some data on Screen 1 and clicks Next.
3. The data submitted by the user is validated on the server.
4. In case of error, the user is returned to the same screen (Screen 1 in this case) with the list of errors.
5. If there is no error, the user is taken to the next screen (Screen 2 in this case).

6. If users click Next on this screen, they are taken to the next screen, if there are no errors. Otherwise, they are taken to the same screen and shown the list of errors.
7. If users click Previous on this screen, they are taken to the previous screen, if there are no errors. Otherwise, they are taken to the same screen and shown the list of errors.

Steps 6-7 are repeated until the user reaches the last screen

Clicking Save on the last screen will save the data to the database and take the user to the Initiate Screen, if there are no errors. Otherwise, the user is prompted to check the data that was entered.

Clicking Cancel on any screen will take the user back to the Initiate Screen after confirmation and all the data entered by the user will be lost.

Please note: If the sequence in which these data entry screens appear is not fixed, then added tab navigation can also be provided which gives a user the flexibility to jump from one screen to another in a random manner.

Table 6 Standard Actions and Navigation

Button	Standard Action	Standard Navigation
PREVIOUS	Data submitted is validated and is stored on the server.	If validation fails the user is returned to the same screen with the errors shown in the Error Block. Else the user is taken to the previous screen.
NEXT	Data submitted is validated and is stored on the server.	If validation fails, the user is returned to the same screen with the errors shown in the Error Block. Else the user is taken to the next screen.
Cancel	Discards any pending changes and navigates off the current page to the Initiate Screen.	Navigate back to the page that initiated the action (Initiate Screen).
Save	Validates the data entered/changed on Screen N, if the data is correctly validated, the entire data saved during the course of Screen 1 to Screen N is committed to the database.	If validation fails the user is returned to Screen N with the errors shown in the Error Block. Else the user is taken to the Initiate Screen.

9.2.6 Master Detail Page

The illustrations used in this section represent the recommended method for execution.

Figure 9 represents the recommended screen flow for a Master-Detail or Parent-Child relationship. To reach a child record, the user must first get the hit list for the parent records and then select the parent record to move on to the child records. The 'Selected Record' block in this diagram is just an indication to the current selected parent record by changing the color of that row. In order to comply with 508 guidelines, indicate through context or markup.

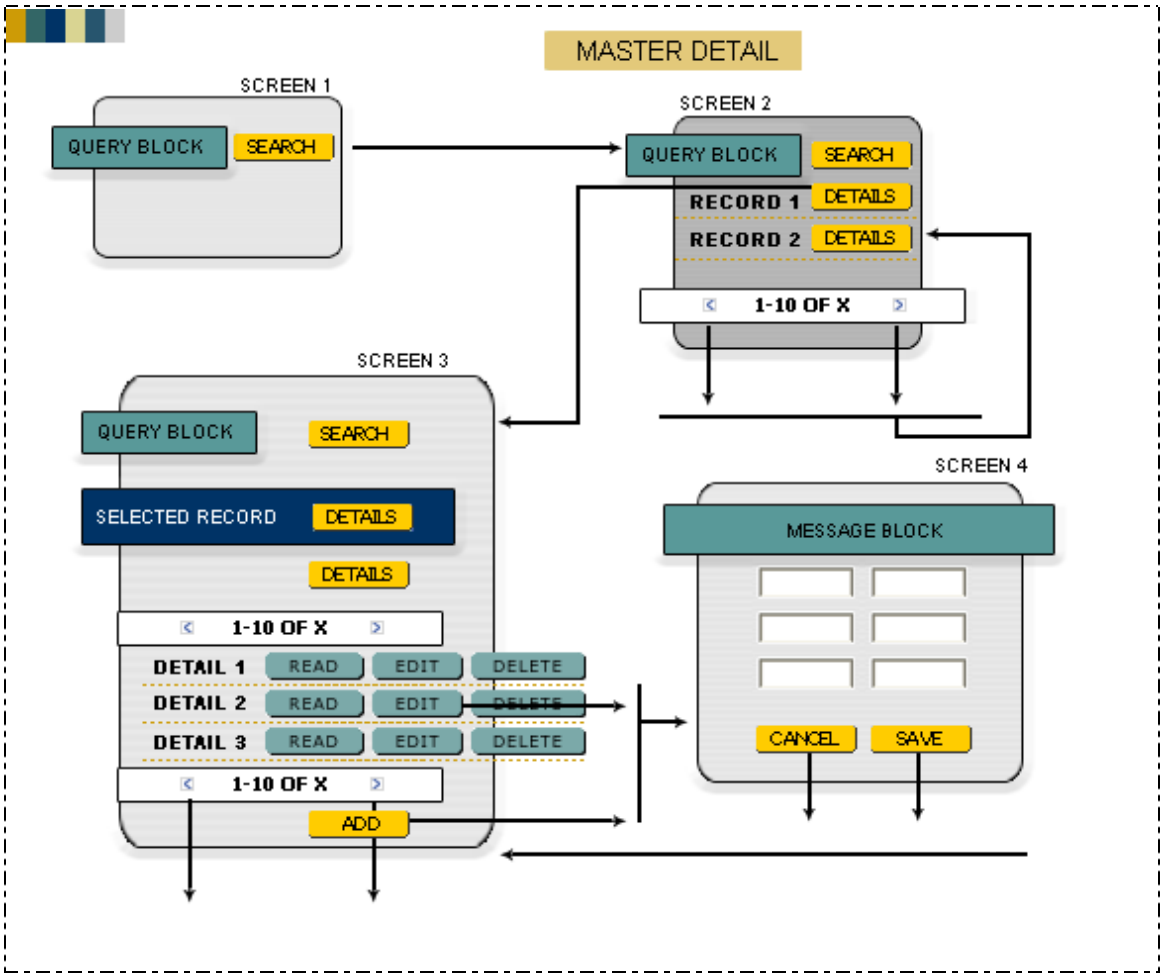


Figure 9 Master Detail Screen Flow

9.2.7 Reverse Master Detail Page

A reverse master/detail page consists of a details section, depicted at top, followed by the master section below. Data will be entered in the details section data entry fields, and choosing Save will cause the data to appear in the master list below. Choosing Edit on one of the master items will cause the data from that item to populate the details data entry fields above. Choosing Delete on one of the listed items will cause that item to be deleted from the master list. Choosing Cancel will work according to the current UI standards.

9.2.8 List of Values (LOV) Page

This flow describes how a non-standard (programmed manually) LOV shall be used in the application. This should also include multi-select LOVs.

1. User clicks the Find button on the application screen.

2. A new popup screen opens up, which contains all the possible values for the required data field. This screen has navigation buttons if the list is large enough to be contained on a single screen.
3. User selects one of the links.
4. The popup window is closed and the value for the selected link is entered in the box adjacent to the Find button.

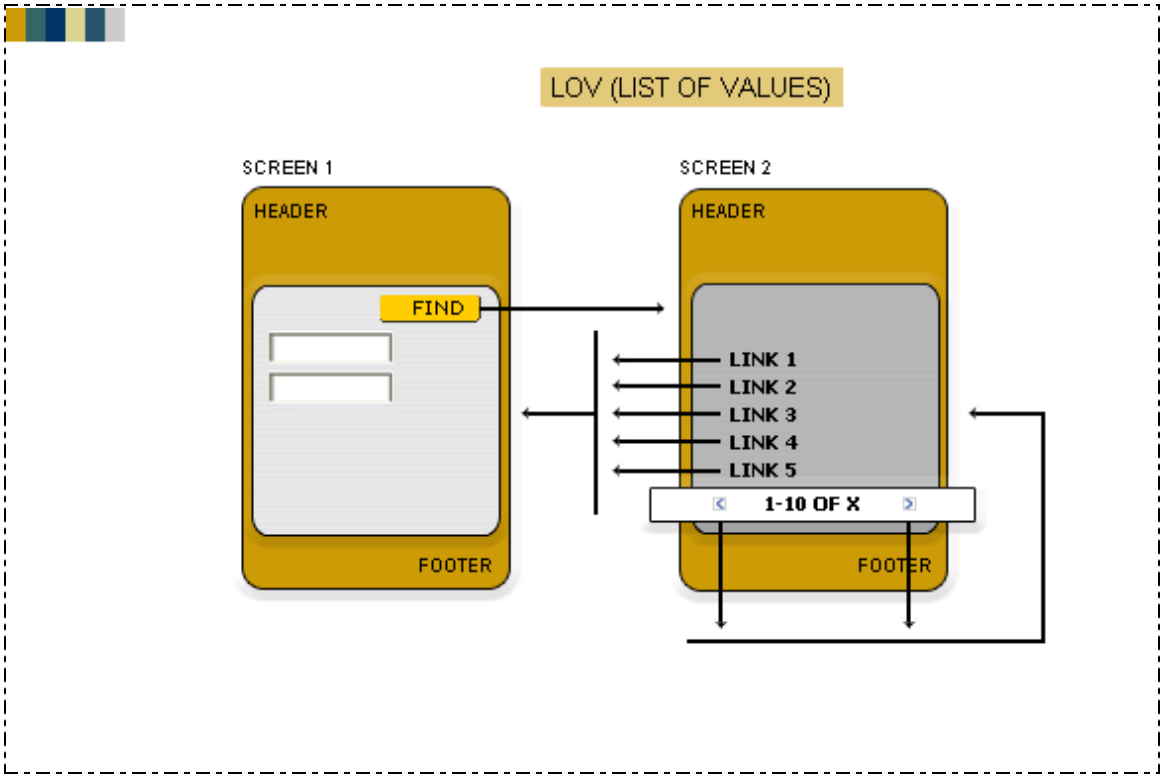


Figure 10 List of Values (LOV) Flow

9.2.9 Import Data Page

This flow is used when there is a need to import data from a file on the user's desk (or other storage. The following is a standard sequence for this flow.

1. User clicks Import File link on the application screen.
2. User is navigated to Screen 1.
3. User selects a file from his/her local machine using the Browse button.
4. After selecting a file, user clicks Import button.
5. User is taken to Screen 2 where all the data from the imported file is displayed.
6. Clicking the Cancel button will take the user to Screen 1 to import the file again.
7. If the user clicks Save on this screen, all the data to be imported is validated.
8. If the data to be imported is not valid, the user is prompted to correct the file and upload the file again using the Cancel button.
9. If the data to be imported is valid, the user is navigated to Screen 3 where a success message is displayed to the user and he is prompted to import more file.

Clicking the Proceed button on Screen 3 will result in a repetition of steps 2-8.

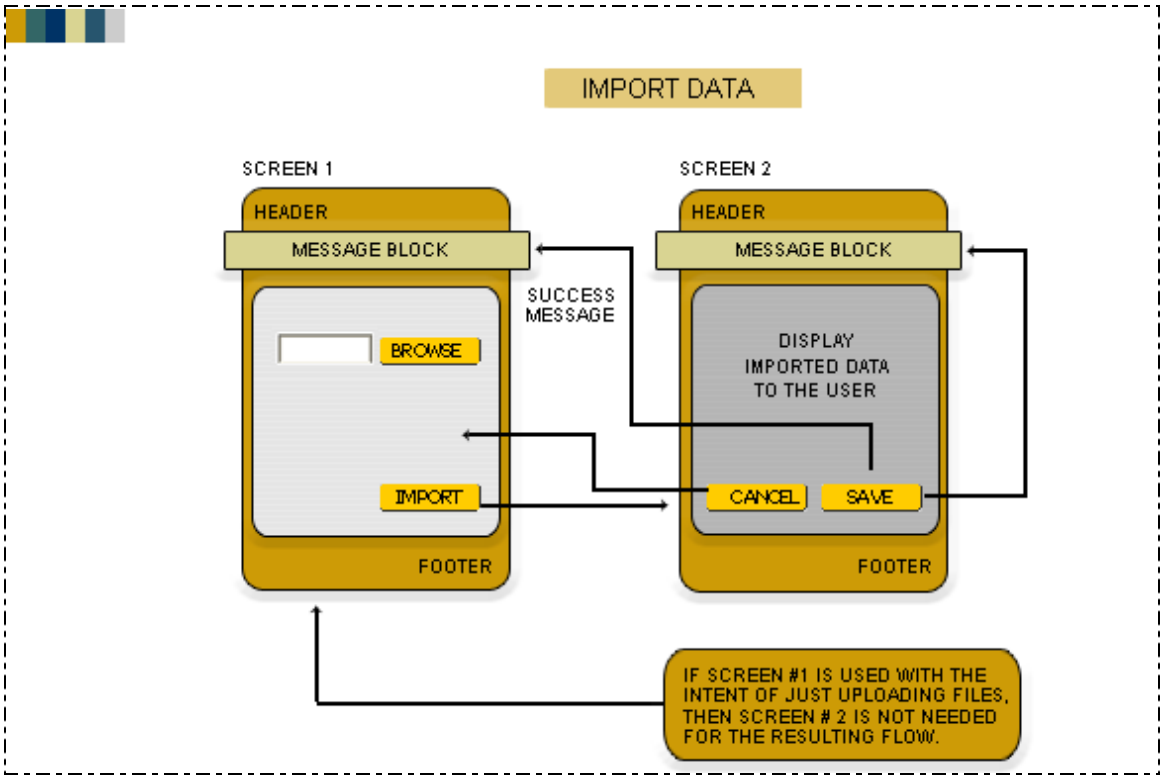


Figure 11 Import Data Flow

9.2.10 Export Data Page

The export flow allows a user to export data from a page or a portion of a page. Typically, a user exports data from a table. There are exporting flow options, depending on whether the data being exported requires a user to fill in extra exporting setting/options prior to exporting the data. Figure 14 Export Data Flow assumes that the data will be exported to an Excel file. Later a drop-down for selecting the file format can be added on Screen 1 when other file formats like PDF, MS Word, etc. will be supported.

1. User clicks Export File link on the application screen.
2. User is navigated to Screen 1.
3. User enters the selection criteria for the data that needs to be exported.
4. After entering the criteria, user clicks the Next button.
5. User is taken to Screen 2 where all the data to be exported is displayed.
   Please note: It may be required to show the data in a specific file format depending on the requirements.
6. Clicking the Cancel button will take the user to Screen 1 to enter the selection criteria again.
7. If the user clicks Export on this screen, he/she will be prompted to save the file on the local machine.

8. After saving the file the user is navigated to Screen 3.
9. Clicking the Proceed button on Screen 3 will result in a repetition of steps 2-8.

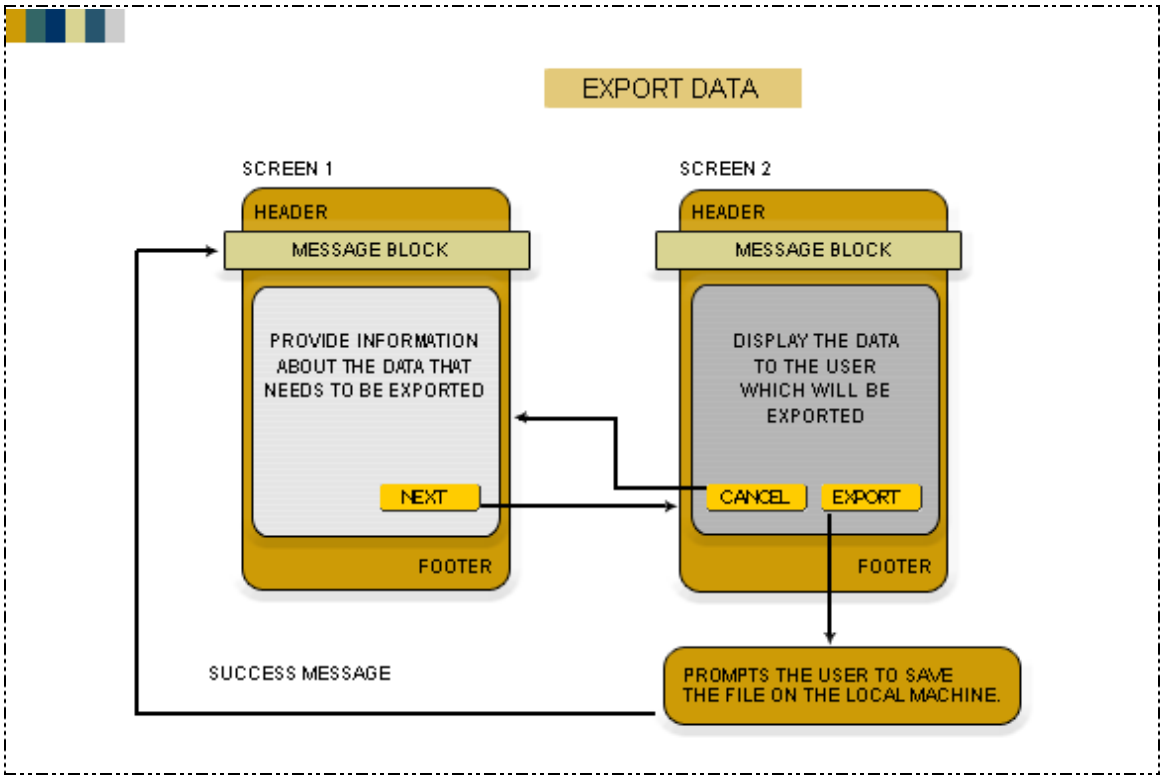


Figure 12 Export Data Flow

9.2.11 File Attachment Page

The attachment flow is used when an attachment or multiple attachments can be associated with an object. Attachments can be added to an object, deleted from an object, or viewed.

9.2.12 Site Map

Site maps should be used in applications that have grouped information from which a user can browse through categories and subcategories of information to get to an object/item list. A user may start a browse flow from the home page of an application. The navigation area of the home page may contain a subset of categories (committees) that the user can select. If the user wants to view the full list, a button labeled "Full List" is selected.

After the selection, the user may be presented with one or many screens depending how deep the information is structured. For instance, a group list (group of committees) may contain meeting schedules, followed by a roster (list or tabular) or it may have a combination of meetings and attendees (list or tabular) on a single page. Once the user finds the item he/she is looking for, he/she may drill down to see the item details.

9.2.13 Contact Us Page Flow

The contact us page flow allows a user to navigate (via a global contact us button) to a global section containing a variety of contact sources. It may contain:

- List of addresses
- Phone numbers/fax numbers
- Emails
- Other contact information

9.2.14 Log on/Log off Page Flow

The log on/log off flow illustrates how a user logs in to an application and/or portal; performs functions, then logs out. Once logged in, the user will have a log off link at the top right side of screen.

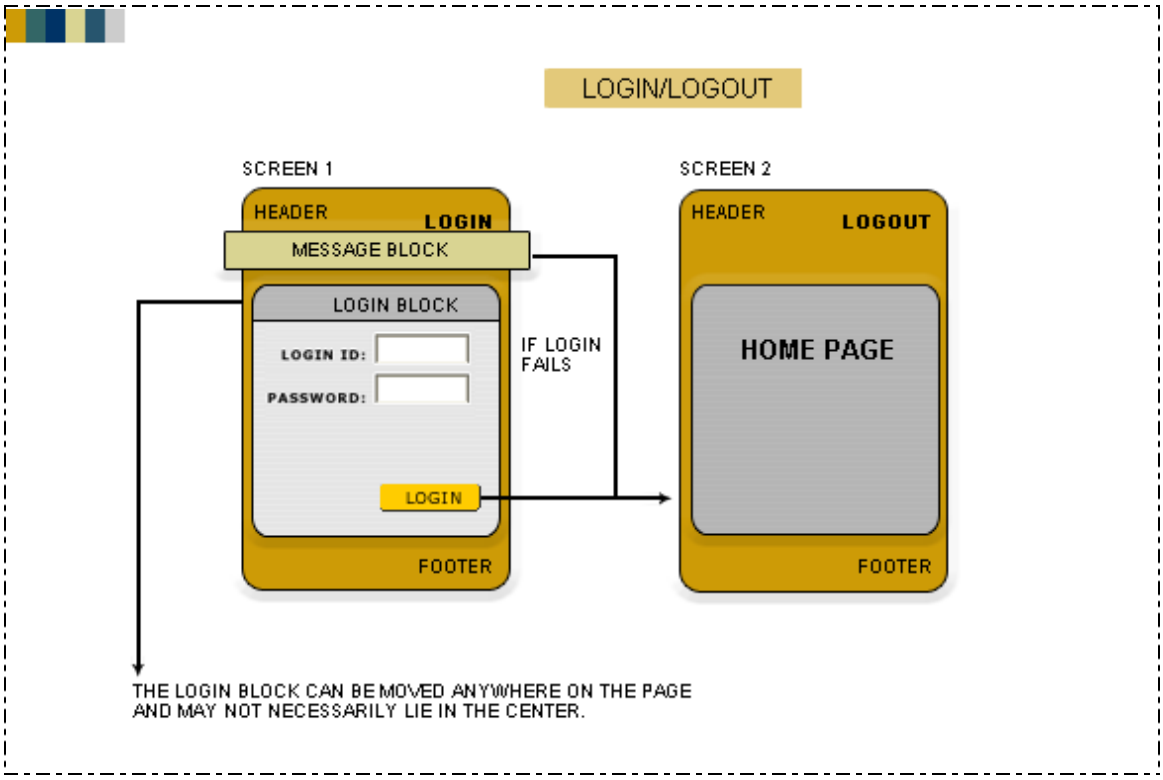


Figure 13 Log on/Log off Flow

9.2.15 Preferences Page

The preferences flow allows a user to navigate via global preferences buttons to a global section to customize his/her UI. The preferences section may contain:

- General preferences for the user for all his/her intranet applications
- Specific preference/customization settings for the application at hand

9.2.16 Printable Page

The printable page flow is used to display a printable page to a user, and allow the user to print the contents of that page. The pages that offer a printer-friendly version of the content will have a button to open the printer-friendly version in a separate popup. A printable page is optional.

9.2.17 Wizard Pages

The wizard-like step-by-step design is used specifically for functionality that had a linear (sequential) process where a task can be presented as a series of steps. Common page flows within wizard pattern:

- Overview/summary/Hit list
- Create/add/delete/update "x"
- Review
- Confirm

9.2.18 Star Pattern

The "star" pattern is used in situations where the application presents the user with a list of objects (possible multiple levels of objects) to select from, and then the user can perform numerous tasks and actions on that object. “Star” relates to list of objects that can take user in many different directions.

Common page flows within star pattern:

- View, browse, objects
- Select object
- View object summary-actions on object summary
- Select object details
- View object details-actions on object details

9.2.19 Pages with Nested Sorts

Nested sorts are required when users need to sort on more than one column in a hit list, or need to sort on more than one component of a column in a hit list, such as the components of the grant number. A maximum of five nested sorts will be provided. If the nested sort requirement is limited to less than five columns, then the number of drop-down lists will be limited to the maximum nested sort requirement. E.g. if the nested sort requirement is for PI Name, Title, and PCC, then only three drop-down lists will be provided.

Nested sorts will be implemented similarly to a standard search. Clicking on the Search button with user defined nested sorts would perform both a search and sort. Clicking on the Clear button will remove both search and sort criteria.

Defaults:

- The nested sort section of the UI will be collapsible. By default, this section will be displayed, but the user will have the option to minimize it.
- On a search screen, the ascending and descending radio buttons will be defaulted to ascending.

- When a hit list is first displayed, the user or application defined default sort criteria will be displayed in the drop-down lists, including the ascending or descending order.
- If the user or application does not select a sort, the developer may define the default sort criteria. The developer chosen sort criteria will be displayed in the drop-down lists, including the ascending or descending order.
- If a user selects the same sort in multiple drop-downs lists, a message indicating that the "sort reference is not valid" will be provided.

10 On-Line Help

Applications shall provide a comprehensive online help system (background information, procedures for specific user instructions, examples, FAQs, etc.) at several levels (see the following three sections). Each page will have a link (icon) to the Page level and Systems level help in the menu bar. These links will launch the independent help system into a separate browser.

Context-sensitive help will have links (icons) to an embedded brief description that will have a link to the more complete topic located in the online (page level) help.
Possible Topic Types include:

- **Task Topics.** Task topics are the numbered steps for doing a task or the procedures to follow. This could also include an example of the output expected when the user has completed a task, such as a fill-in form or a dialog box with the correct options selected.
- **Context Topic.** Context Topic describes an item in the application interface that is accessed by the user pressing keys or clicking an item in a prescribed way.
- **Reference Topic.** A reference topic provides either non-procedural information or an explanation of concepts, ideas, theories, and methods. It may also include listings, such as a directory of commands in the application.
- **Lookup Topic.** Values, variables, parameters, or other data needed.
- **Definition Topic.** Defines a term.
- **Navigation Topic.** Navigation Topics make up a list of jumps to help the user navigate through an online document.

10.1 Alert Help

This shall be used to call attention to useful domain or UI information that might not otherwise be obvious. This information may include reminders about organization policies, UI shortcuts or alternate ways to perform tasks, pointers to Preferences, or information on how the UI behaves, such as "Changes will be saved when you move to the next <Committee>."

10.2 Context-Sensitive Help

Context-Sensitive Help is a method for providing contextual help as a short snippet of information (e.g., brief description) to assist the user, without forcing them into a formal help system. The application shall provide context-sensitive help (instructions, examples, etc.) for user input fields or groups of fields where the field is not self-explanatory. All context-sensitive help shall be identified using the icon, which contains an ALT tag with the label “Help.”

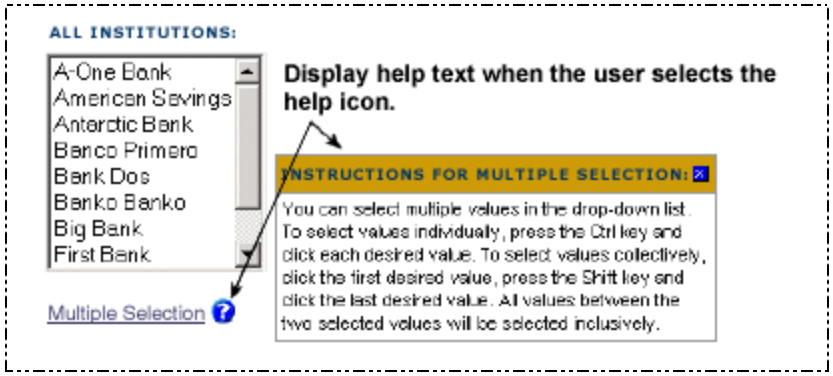


Figure 14 Context Sensitive Help

Typically, context-sensitive help will contain:

- Brief information about the section or control that the help was associated with. The content of context-sensitive help will be developed based on the complexity of the specific field (i.e., the Username field might indicate that a username is case-sensitive).
- Links to related page level help, anchored to the appropriate section (if applicable)

10.2.1 Implementation Style

This type of information is frequently implemented via html layers, as this provides "quick" access for the user. The layer can render immediately, without having to wait on the performance delay of a round-trip to the server to display the help information. Of course the flip side of that is ensuring that these snippets are used judiciously so the HTML page is not thickened with snippets. There is also the accessibility issue that comes with layers, but that issue will have to be dealt with in other areas (e.g., cascading navigation, etc).

Using links within these contextual help layers (to FAQ's or Page level help) will help ensure that the snippets remain a manageable size, but they also allow for easy navigation to additional help resources. This supports the judicious and selective use of context-sensitive help, rather than use it on EVERY field, since the complete help index will be available easily.

10.3 Context-Sensitive FAQs

As a feature of the context-sensitive help, the user will have access to a list of Frequently Asked Questions (FAQ) that are specific to the topic for which the user requested help.

10.4 Page-Level Help

A separate help page will be available for each page of the application. Page level help should be the primary location for DETAILED information about specific tasks and functions on a particular page. These areas should serve as the destination link from the contextual help snippets (e.g., anchor tags within the page level help). Access to page level help will be through the help icon in the menu bar. The general structure of the page level help will be as follows. Detailed overview and procedural information about each major section on the page, which may also include detailed description of specific sections or controls, as needed. This additional detail will serve as the target for context-sensitive help's [More…] link. Links to:

- System Help Home Page
- Table of Contents
- Help Index
- FAQ (or links to FAQs that are relevant to that specific page)
- Page level help will also contain links to the System Level Help home page, FAQ's (or FAQ's for this page), Index, etc.

10.5 System-Level Help

Similar to page level help, system level help pages will be available. Access to system level help will be through the help icon in the menu bar, as well as links from page level help. The general structure of the System-Level help will be as follows. A comprehensive overview of the application, with high-level descriptions for the system links to:

- A detailed table of contents for all of the help system (link to context-sensitive help)
- Table of Contents for the FAQ system
- Help Index
- Glossary (if applicable)

10.6 Hover Help

It is strongly recommended that all applications shall display descriptive information whenever the mouse icon is place over or hovers over an image within the application. Typically this is accomplished with an image's ALT tag, if some other mechanism is utilized to generate this hover help, then appropriate measures will have to be taken to include an accessible alternative.

10.7 Descriptive Text

The applications shall provide examples of the input being requested from the user wherever there is a possibility of choosing different formats. Examples are dates, abbreviations, etc. Care should be taken to ensure consistency across the applications for similar inputs. For example, do not allow date input to be gathered as mm/dd/yyyy in one area but dd/mm/yy in another area.

10.8 Application Support

10.8.1 Screen Ids

All pages include this important feature contain a rendering DTG and screen identifier. This information should be contained on a hidden layer, which has some obscure, but easily accessible activation link that can be communicated by NCICB Application Support personnel.

10.8.2 E-Mail Link

All pages shall contain the capability for the user to email the NCICB Application Support. When invoked, the Subject line of the email shall be pre-loaded with the user's current screen id.

11 NCICB Portal Behaviors (TBD)

Define standard portal behaviors.
Define page flow within portal and launching outside the portal.
Define flow for integration with external and internal applications.
Define standards look and feel for portals.

12 Message / Feedback Text

There are various types of messages and feedback text that can be created by an application and various forms in which this information will be communicated to the users. Each has a specific purpose and should be used appropriately. There are five types of messages supported by this standard.

12.1.1 Types of messages:

- **Information** - Provides contextual information about business practices and methods.
- **Warning/Alert** - Alerts the user to a condition or a situation that requires a decision before continuing.
- **Error** - Alerts the user that one or more actions have failed and the application requires immediate attention and/or correction before continuing.
- **Confirmation** - Notifies the user that an action has been completed successfully.
- **Processing** - Notifies the user that an action is currently being processed.

Each of these types of messages provides feedback of some sort to a user about a condition, situation, or circumstance of concern. The severity of the message is reflected in the content of the text itself. The display of the message can take on various forms depending upon the application, the requirements of the users, and the situation under which the message arose. The display formats that are supported by this standard include:

- **Inline Messages:** These are included right on the page and primarily used for information. Typically, such it is a one line message about small section or a widget on the page and is used to guide users. However it can be used to provide warnings and error information.
- **Message Boxes:** They are used at the page level, primarily as a confirmation of an action, warning and processing message.
- **Message Pages:** Typically they are used in response to a processing request. They provide information related success, failure, processing and next steps. They typically have a button(s) for user to navigate further. The combination of the types of messages and the display formats provides for every conceivable presentation needed for the project.

12.2 Error Messages

Error messages should present a clear visual alert to the user that a problem has occurred. In addition, the text of the message should contain an error identifier, describe the problem precisely, and recommend a course of action to solve the problem. Error messages should contain a (n):

- Explicit indication that something has gone wrong. Clarify the error by supplying a specific message and indicate the matter in which the error can be corrected. (Example: *"Your email address was not formatted correctly."*)
- Human-readable language, instead of obscure codes or abbreviations such as *"a type 2 error has occurred."*

- Precise descriptions of the exact problems, rather than vague generalities such as "*syntax error*."
- If relevant, include an error Id number (at the end of the error message) to assist NCICB Application Support personnel in determining and correcting the problem.

If the error message is presented in a separate message box (e.g., JavaScript Alert box), it should contain a meaningful title, to inform the user which application caused the error.

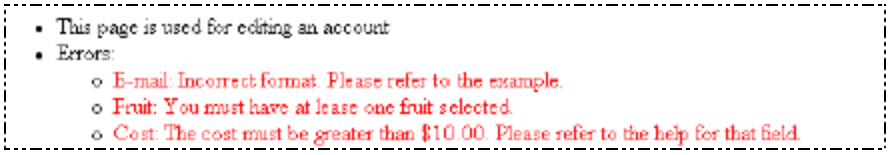


Figure 15 Error Message

Please note: Avoid phrasing that blames the user or implies that the user is to blame for the error. Avoid the use of exclamation points in the error message text.

Properties for Error Message Pages:

- Identify the action that failed.
- Use project standard error numbers.
- If it is a validation error provide directions to resolve it, and optionally append a back-end error code.
- If it is a system type error, provide a link to dump system information for diagnostic purposes. On a "dump" page include a "mailto" link to the NCICB Application Support.

12.3 Message Boxes

Message boxes are mechanisms to provide users with additional information without changing the screen context. Alert messages and confirmation messages are two typical message boxes used in the web environment. Ensure that the message text is clear, concise, and in terms that the user will understand. Avoid technical jargon, system-oriented information, error codes, etc. Avoid phrasing that blames the user or implies user error.

Typically, these messages are generated through the use of JavaScript code. If this is the case, care must be taken to ensure that critical information or functionality is not denied to those users who disable JavaScript in their browsers (e.g., those using screen readers, etc.). Typical solutions for users of assistive technology include:

- Omit message boxes completely, and instead present information as separate HTML pages.
- Provide assistive technology links (e.g., hidden d-links) next to the Message box trigger link. The d-links would navigate the user to an HTML message page, with identical information and options as the message box.
- Provide a non-script tag that allows the user to redirect to a non-JavaScript-enabled page.

Figure 16 Message Box

13 Special Considerations

13.1 Design Considerations

- Avoid horizontal scrolling with a scroll bar. Use a [More...] button or Tab Pages to display additional screens of data in one refresh. Use Tab Pages to cluster alike data on separate pages to improve navigation and display.
- Use vertical scroll bars and page up and page down to scroll up and scroll down sets of records at a time.
- Designing forms for the web requires a judicious balance between the size of the overall form and calling new forms. It also requires care in navigation as painting new windows will take more time than in client server.
- Minimize the movements required to switch base screens. Users should be able to return directly to the main menu to choose new screens. When leaving screens, transactions will be committed or discarded.
- The menu will change dynamically with each major option but should not change for windows and pop-ups. Options are grayed out where not available.
- Menu choices direct navigation and serve as notification of decision. When a new option/window is selected, the current transaction/change is committed automatically with no alert. In any business area, users can request an alert if the decision is critical. But the default is not to provide alerts, simply save when the window changes.

13.2 Multimedia Presentations

To avoid accessibility and performance problems, it recommended that the use of multimedia presentations in your applications be avoided unless absolutely necessary. The use of SVG technology is supported as well as media that requires Real Player; however Flash and Shockwave are discouraged.

Multimedia presentations include audio and/or video of conferences, speeches, introductions, trainings, etc. These are generally not relevant to the function of our applications and should not be used. Functionality that requires the use of external applications should not be used because those applications are rarely accessible to users that require assistive technology. We are to assume that in any case where it is necessary for the user to download and install an application, the user may not have access to the application, or may not have the skill or experience with downloading software that would be required to install the application. For this reason multimedia supported by NCICB and its entire internet entities should be limited.

13.3 Scripts

While applications may utilize JavaScript to accomplish specific functions, the design must provide script information in a fashion that can be read by assistive technologies. When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology. Assistive technologies and alternate browsers in most cases cannot support scripts, applets, or plug-ins, causing the content rendered in these ways to be completely inaccessible. Additionally, some users prefer or are required to disable the use of scripts, applets, and plug-ins to maximize system security. In general, design your Java applications to be capable of

interaction with assistive technologies. Wherever scripts are used, include the NOSCRIPT tag to describe the activity that the scripting enables.

Example:
Whenever a script is added to a document, a NOSCRIPT tag should also be included:
<SCRIPT type="text/java script">
... display and replace each frame of a comic strip as requested by a user event ...
</SCRIPT>

<NOSCRIPT>
In the first panel, Lucy is holding the football. In the second panel, Charlie Brown is seen charging towards the football. And so on..."
</NOSCRIPT>

13.4 Applets and Plug-ins

Applets and plug-ins provide additional functionalities to HTML for such features as multimedia presentations, including audio and video presentations, and for graphics-based technologies such as Flash and Shockwave. These are generally not relevant in our applications and should be avoided. When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).

Functionalities requiring the use of external applications are rarely accessible to users with assistive technologies. Additionally, because the applets or plug-ins are often proprietary software owned by private companies, they may or may not be included in a standard computer configuration. If they are not, it is necessary for the user or their system administrator to download and install the application. Many users may not have access to the application, or may not have the skill or experience with downloading software that would be required to install the application. To avoid accessibility problems, the application shall avoid using these software packages.

13.5 Electronic Forms

When electronic forms are designed for completion on-line, the forms shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues. Any application form designed for completion online will allow users with assistive technology to access the information, field elements, help, and functionality required for completion and submission of such form, including any directions or cues.

Currently, the interaction between form controls and screen readers can be unpredictable, depending on the design of the page. HTML forms post accessibility problems when web developers separate a form element from its associated label or title, making it difficult for users with assistive technology to provide the correct information in the correct field.

Developers shall design electronic forms for completion online so that assistive technologies such as screen readers can relay the relevant field descriptors when tabbing from field to field. A logical tabbing order must be used consistently throughout the form for all data entry fields.

Labels will be associated with all form elements. The label will appear either immediately to the left of immediately above the field. Additionally, all form controls, including checkboxes and radio buttons, will have text descriptions.

Example:
If an input box is intended to receive the user's last name, the developer must be careful that the text label for this field, "Last Name", appears near that input box or is associated with it. In many cases, developers list all labels in one column, then all fields in another column, making the completion of this form very difficult for unsighted users who cannot visualize the placement of the labels. For example, one could design a form that asked the following:

****
SUBMIT
What is your favorite color?
Red [checkbox]
Green [checkbox]
Blue [checkbox]

Company Name

RESET

Skip the form to find accessible alternatives.
****

To make the above form accessible and understandable to a user with a screen reader or keyboard-only input with the following code:

```
<FORM METHOD="POST">
<INPUT TYPE="SUBMIT" VALUE="Enter" tabindex="6">
What's your favorite color? <BR>
Red <INPUT TYPE="CHECKBOX" NAME="CBVar1" VALUE="CB4"
tabindex="3"><BR>
Green <INPUT TYPE="CHECKBOX" NAME="CBVar2" VALUE="CB5"
tabindex="4"><BR>
Blue <INPUT TYPE="CHECKBOX" NAME="CBVar3" VALUE="CkB6"
tabindex="5"><BR>
Company Name <INPUT TYPE="TEXT" NAME="Com" tabindex="2"><BR>
<INPUT TYPE="RESET" VALUE="Undo" tabindex="7">
<A HREF="form-alt.htm" tabindex="1">Skip the form to find accessible
alternatives</A>.<BR>
</FORM>
```

Please note: The use of the "tabindex" attribute, in the above code, allows a programmer to define the order in which actions in a form occur for screen reader or keyboard-only users.

13.6 Performance

13.6.1 Graphics

Visual effects are appealing, but can be costly. It is recommended that the developers avoid graphics that cause significant performance degradation. Also, for accessibility considerations, do not use of graphics alone to convey information.

13.6.2 Multimedia Objects

Flash and other multimedia "plug-in" delay the users' ability to get what they came for. It is recommended that multimedia effects be avoided unless it is relevant to the specific feature.

13.6.3 Time Delays

Users with physical or cognitive disabilities may be slower than the general public to enter data or complete a form within a given time allotted. Time delay pages must alert users that data needs to be entered, and give users the option of extending the time necessary to complete the form information. The application shall not include any scripts that will cause a web page to disappear, "time out," or expire if a response is not received within a specified time. When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required (e.g., users should have the option to extend the time they require to complete the form).

Appendix

A. Acronyms and Definitions

The terms, definitions and roles listed below are provided as an aid in understanding the application of requirements management principles and processes.

Acronyms

CRUD	Create, Read, Update, and Delete
CSS	Cascading Style Sheets
DHTML	Dynamic Hyper Text Mark-up Language
GIF	Graphics Interchange Format
HHS	Department of Health and Human Services
HTML	Hyper Text Mark-up Language
JPEG	Joint Photographic Experts Group
LOV	List of Values
NCI	National Cancer Institute
NCICB	National Cancer Institute Center for Bio-informatics
NIH	National Institute of Health
SVG	Scalable Vector Graphic
UI	User Interface
URL	Uniform Resource Locator
XML	Extensible Mark-up Language
XSL	Extensible Style sheet Language
XSMLT	Extensible Style sheet Language Transformations

Definitions

508 Compliance – 508 Compliance is the law that requires all government agencies to provide individuals with disabilities the means necessary to use and access all electronic information. Please visit *www.section508.gov* to learn more about 508.

JavaScript – JavaScript is a simplified C-like syntax that is tightly integrated with the browser Document Object Model. It is useful for implementing enhanced forms, simple web database front-ends, and navigation enhancements.

Shockwave – Shockwave is a Macromedia product used to view files created in Macromedia Director.

Breadcrumbs – Breadcrumbs, also known as a breadcrumb trail, is a navigation tool that allows a user to see where the current page is in relation to the website's hierarchy. Please visit *http://www.welie.com/patterns/crumbs.html* to read more about breadcrumbs.

B. Overview of Style Sheets

There are two leading types of style sheets: cascading style sheets (CSS) and Extensible Style sheet Language (XSL). The use and justification for two different Style Sheet languages is found in the table below.

	CSS	XSL	Related Notes
Can be used with HTML?	Yes	No	While CSS can be used to style HTML and XML documents, XSL transforms documents. For example, XSL can be used to transform XML data into HTML/CSS documents on the web server. This allows the two languages to complement each other and be used together.
Can be used with XML?	Yes	Yes	Both languages can be used to style XML documents.
Transformation language	No	Yes	CSS and XSL will use the same underlying formatting model and designers will therefore have access to the same formatting features in both languages.

Cascading Style Sheets (CSS)

Cascading style sheets allow designers to capture a separation of presentation and content. The web is a true cross-platform system and content is presented on such a wide variety of devices that pages should specify the meaning of the information and leave presentation details to a merger (or "cascade") of site-specified style sheets and the user's preferences. This makes it far easier for a designer to introduce new page designs by simply modifying a single style sheet or creating a single style sheet rather than by modifying thousands of content pages.

One of the main benefits of style sheets is to ensure visual continuity as the user navigates your site. Printed publications have long known the value of basing print products on a single typeface: no matter where you turn in a magazine or a newspaper, the text and basic layout looks the same. Websites will gain the same brand cohesiveness if all the pages on a site link to the same style sheet.

Syntax Overview

Before defining the features, functions, and application of Cascading Style Sheets for this standard, a brief overview of the syntax of the style sheet commands may prove helpful. Every CSS, whether it is contained in its own file or embedded in the head element of an HTML document, is composed of a series of instructions called *Statements*, which do two things:

- **Identify the elements in an HTML document that it affects.** In HTML terminology, an element is anything marked up inside HTML tags, including page elements. That portion of a statement that identifies page elements is called a Selector, used to select page elements.
- **Instruct the browser how to draw these elements (e.g., paragraphs, links, list items, etc.).** The part of a statement that instructs a browser how selected elements should be

drawn is called the *Declaration*. A declaration element can contain quite a number of *Properties* (individual style pieces that are to be applied to the selected).

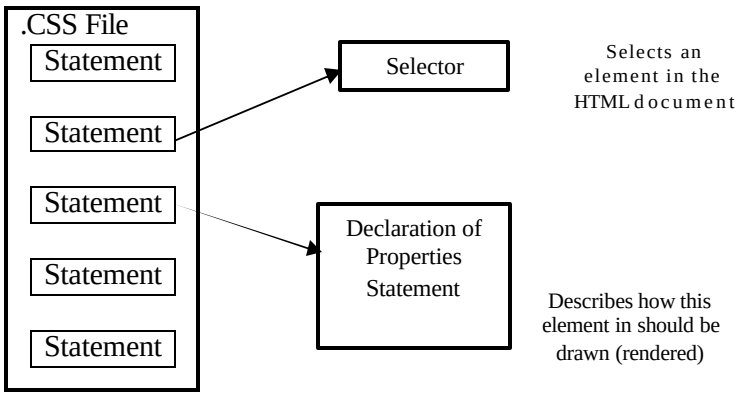


Figure 17 Syntax Overview

Example:
Body {font-family: Verdana, "Minion Web", Helvetica, sans-serif; font-size: 1em; text-align:justify;}

This is a single statement, perhaps one of many in a style sheet. In this example, the selector is body, so this statement will affect the <body> element of any page linked to this style sheet. The statement has a declaration with three properties, so chosen by this selector will have three of its properties affected. Each property has a name and a value, for example "text-align" and "justify", respectively. In this case, the font of the body text will be drawn in Verdana by the browser. If Verdana is not available, the browser will choose the next available font; e.g., Minion Web, then Helvetica if that one is not available, and so forth. If none of the specified fonts are available, the browser will draw the body text in a default sans-serif font.

This statement also sets the font size to one (1) em. Since in HTML you cannot set the font point size, only choose one of six relative font sizes, this is another distinct advantage of style sheets - much more sophisticated page layout and typographical control.

Style Sheet Standards

Understanding the basic syntax of style sheets is essential to coding them properly for use in applications. Browsers are not inherently tolerant of deviations in syntax, and either your style sheet will not work at all, or it will work in unexpected ways. Using the above example as a reference, the following are some rules to follow when creating a style sheet.

- Every statement must have a selector and a declaration.
- The declaration appears immediately after the selector and is contained by a pair of curly braces.
- The declaration is one or more properties separated by semicolons.
- Each property has a property name followed by a colon and then the value for that property. There are many types of values, but a property can only take certain values as set down in the specification.
- Sometimes a property can take a number of values, as in the “font-family” example above. A comma and a space should separate each value in the list.
- Sometimes a value will have a unit as well as the actual value, as in the “font-size” example above. Do not put a space between the value and its unit.
- As with HTML, white space can be used to make your style sheet easier to read and write. Use spaces as needed to improve readability and maintainability.

Implementation Advice

This section contains some tips on how to implement style sheets most effectively.

Use either a single style sheet for all of the pages on the site or a few coordinated ones if there are pages with very different needs.

Always use linked style sheets rather than embedded styles. Referencing an external file provides the maintenance benefits of being able to update the look of your entire site with a single change. Also, by pulling style definitions out of your pages, you make them smaller and thus faster to download. If you use a single style sheet for your entire site, that file will be a single download once and for all.

- If many pages need the same effect, it should be added to the site's global style sheet.
- Single-page styles should be embedded rather than linked: the page should continue to link in the global style sheet and then override it with local, embedded styles as necessary. Doing so has the benefit of allowing future changes to the central style sheet to propagate to the modified page to the greatest extent possible. Pages must continue to operate even when style sheets are disabled. Retaining a decent presentation without the style sheet is mandatory to support users with older browsers, Section 508 compliance, and any users who choose to disable the style feather in their browser (either because of bugs or because your style conflicts too much with their preferences). It is very easy to check conformance with this rule: simply disable style sheets in your browser and reload the page.
- Do not use more than two fonts. Using a lot of fonts results in an unprofessional look. Typically, you use one typeface for body text and another, bolder, face for headings. *Please note: It is recommended you use a long list of alternate fonts in the style sheet specification for a given class of text. The user's browser will pick the first available font in the list and use it throughout your pages, meaning that the user will see a single font, making the site feel typographically unified. It is important that lists of font names have the fonts listed in the same order, since the browser picks the first one it has available.*
- Use ems to specify your font sizes.
- Specify all text relative to the base font size defined by the user's preference setting. For example, large text could be defined as "200%", meaning that it would be set as 24 point

if the user preferred 12 point for body text. Whether people prefer large or small fonts depends on a variety of questions, including the size and resolution of their monitors and their eyesight. It is annoying to visit a website where the text is too small for comfortable reading and you cannot enlarge it because the font sizes were defined as an absolute number of points.

- Do not use the "important" attribute to override the user's settings. It is hard to imagine cases where you are justified in ignoring the user's preferences if they felt strongly enough to use their own "important" rating, so "important" should be reserved for user style sheets.
- If you have multiple style sheets, make sure to use the same CLASS names for the same concept in all of the style sheets. Content creators using two or more style sheets will be confused if different Classes are used for the same thing or if one style sheet has a CLASS that is missing in the other style sheet even though the concept applied in both cases. If, for example, you have a CLASS for the name of the author of a document, then all of your style sheets should have this CLASS, even though it may be defined to render differently, as appropriate for the different kinds of documents.

A Cascading Style Sheet is just a simple text file; but unlike an HTML document, you do not need a special declaration at the top of the file to say that this is a style sheet. The name of the file should end with a .CSS suffix.

Linking and Embedding a Style Sheet

The style sheet contains the instructions to a browser detailing how to draw a particular page. But for the browser to use the style sheet, it must know the location of the style sheet and when to use it. There are two ways to associate a style sheet with an HTML document.

Embedding Style Sheets

Style sheets can be embedded into the head element of HTML documents, but this is not always the best approach. If the style sheet is embedded into the web page, then the ability to change the appearance of a website by implementing changes to a single file (the style sheet) will be lost. The goal of separating appearance from content will not be achieved.

To implement such a style sheet though, the style sheet itself is placed between a style tag as follows: <style type="text/css"> </style>

Embedding is commonly used by applications, such as Dreamweaver, that make use of the page layout aspects of CSS.

Linking to Style Sheets

The preferred method of associating web pages with style sheets is to place a link in the head of the HTML file to the style sheet. With this link, when the browser starts reading the page, it sees the style sheet link, downloads the style sheet, and then uses it to draw the page. You can place links to several style sheets in an HTML file. This does not mean that a browser uses each of them. Rather, the browser chooses one of these as the style sheet to use (theoretically, the browser should ask the user which style sheet to use).

In this approach, the web page links to the style sheet, yet the style sheet has no knowledge of the pages that are linked to it. To link a web page (HTML document) to a style sheet, you place a link to the style sheet in the head of the document, using the following syntax:
<link rel="stylesheet" type="text/css" href="http://www.xyz.com/style/style.css" />

The attributes are:

rel="stylesheet" indicates that this is a forward link, and the browser should expect a style sheet at the other end.

type="text/css" tells the browser what format it is going to receive the style sheet in. This attribute indicates that this style sheet is written in Cascading Style Sheets (CSS). Theoretically, style sheets can be written in any number of languages.

Generally, we use CSS; however, Extensible Style Language (XSL) is another that may become important. The content type is necessary.

href="http://www.xyz.com/style/style.css" identifies the storage location of the style sheet. This can be either a relative or absolute URL. Remember, the relative URL is relative to the HTML document, unlike URLs within style sheets that are relative to the style sheet.

Note that the " />" closing of this element is the correct way of closing an empty element in XHTML 1.0.

Background color css definitions	.clsBlackBack { background-color: black}
General css text definitions	.clsDefault {font-family:Arial,Helvetica,sans-serif; font-size :12px; color :Black}
Tab/navigation css definitions	.clsMainMenu {background-color: number003366; text-align:center}
Header css definitions	.clsTableHeader { font-family:Arial,Helvetica,sans-serif; font-size:12px; font-weight:bold; background-color: numberCCCCCC; textalign: center }
Table css definitions	clsTable { padding-left: 0px; padding-right: 0px; width: 100%}

Extensible Style sheet Language (XSL)

XSL is a newer language for expressing style sheets. XSL uses a XML notation, whereas CSS uses its own. In CSS, the formatting object tree is almost the same as the source tree, and inheritance of formatting properties is on the source tree. In XSL, the formatting object tree can be radically different from the source tree, and inheritance of formatting properties is on the formatting object tree.

A common confusion with these style sheet languages is that you should use CSS for HTML and XSL for XML. However, CSS can be used with HTML and also with XML, provided that the XML document has a reasonably linear structure that can be displayed without extensive manipulation. XSL is targeted at XML, in particular highly structured, data-rich documents that require extensive formatting. Aside from these technical differences, the primary different is support. CSS1 and (parts of) CSS2 are readily available and supported, while XSL is still too new to have mature browser and content-authoring support.

An XSL style sheet is, like with CSS, a file that describes how to display an XML document of a given type. XSL shares the functionality and is compatible with CSS2 (although it uses a different syntax). Originally intended to perform complex styling operations (e.g., tables of contents and indexes), it is also used as a general purpose XML processing language.

An XSL Style Sheet consists of three parts:

1. XSL Transformations (XSLT): a language for transforming XML documents that is widely used for purposes other than XSL (e.g., generating HTML web pages from XML data).
2. XML Path Language (XPath), an expression language used by XSLT to access or refer to parts of an XML document.
3. XSL Formatting Objects: an XML vocabulary for specifying formatting semantics and advanced styling features, expressed by an XML document type that defines a set of elements called Formatting Objects, and attributes (in part borrowed from CSS2 properties and adding more complex ones).

An XSL style sheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary.

Style Sheet Considerations

While style sheets can significantly improve the consistency, appearance, and use of web pages while reducing development time, they are only useful if users let their browsers use them when constructing screens. Style sheets are particularly useful in selecting specific fonts, controlling the size of fonts to improve readability, and providing user-selected colors for optimum color contrast. The visually impaired often employ the use of their own style sheets that are customized to provide the best possible displays to assist them in reading the web pages or assist their screen readers in interpreting the web pages.

Because visually impaired users often turn the style sheet use off in their browser configuration and opt to use their own style sheet instead, web pages must be organized so that they are also readable without requiring an associated style sheet. In addition, designers must ensure that web pages do not interfere with user-defined style sheets. If not designed properly, style sheets associated with a web page can make accessing that web page difficult for users who have disabled support for style sheets, or for those who use customized style sheets that override the supplied style sheet.

Style commands will be arranged so that the content makes sense and reads in the correct order without any associated style sheet. All web pages will be usable when the style sheets have been disabled in the browser or the user has elected to activate their user-developed style sheet.

Example:

```
<STYLE TYPE="text/css">
.part1 /* The quick */ { padding-left: 0;
color: red; font-size: 14pt;
font-family: copperplate gothic bold, fantasy, sans-serif }
.part2 /* brown fox */ { padding-left: 100px;color: brown; font-size: 10pt;
font-family: times new roman, desdemona, serif }
.part3 /* jumped over */ { padding-left: 350px;
color: purple; font-size: 18pt;
font-family: desdemona, times new roman, serif }
.part4 /* the lazy dog */ { padding-left: 350px;
color: blue; font-size: 24pt;
font-family: fantasy, copperplate gothic bold, sans serif }
</STYLE>
<DIV class=part4>the lazy dog.</DIV>
<DIV class=part1>The quick</DIV>
<DIV class=part3>jumped over</DIV>
<DIV class=part2>brown fox</DIV>
```

In this example, if a user elected to use their own style sheets or if style sheets were disabled in the browser, the sentence would read incorrectly:

```
the lazy dog.
The quick
jumped over
brown fox
```

Example:

```
<STYLE TYPE ="text/css">
<!--
.part1 /* The quick */ { color: red; font-size: 14pt;
padding-left: 0; margin-top: 40px;
font-family: copperplate gothic bold, fantasy, sans-serif }
.part2 /* brown fox */ { color: brown; font-size: 10 pt;
padding-left: 100px; margin-top: 30px;
font-family: times new roman, desdemona, serif }
.part3 /* jumped over */ { color: purple; font-size: 18pt;
padding-left: 200px; margin-top: -60px;
font-family: desdemona, times new roman, serif }
.part4 /* the lazy dog */ { color: blue; font-size: 24pt;
padding-left: 350px;
margin-top: -100px; margin-bottom: 100px;
font-family: fantasy, copperplate gothic bold, sans-serif }
```

```
-->
</STYLE>
<DIV class=part1>The quick</DIV>
<DIV class=part2>brown fox</DIV>
<DIV class=part3>jumped over</DIV>
<DIV class=part4>the lazy dog.</DIV>
```

In this example, if a user elected to use their own style sheets or if style sheets were disabled in the browser, the sentence would read correctly:

The quick
brown fox
jumped over
the lazy dog.

C. HTML Frames

Frames Versus CSS

To further understand the methods available for bypassing the use of frames, it might prove helpful to compare Cascading Style Sheets (CSS) with frames. CSS is an elegantly designed extension, whereas frames create more difficulty than benefit. Specific differences include:

- CSS is backward compatible so that viewing a style-enhanced site with an older browser causes no problems. Of course, the user will not see the stylistic enhancements made possible by CSS (e.g., multiple fonts and indented margins), but the text of the page will be readable and presented in a reasonable default style. In contrast, a page designed with frames is useless for a user with an old browser.

- CSS integrates closely to other features in web browsers. When multiple style sheets become supported in future releases of the mainstream browsers, users might want to learn the command to switch between styles, but they won't have to. In contrast, frames destroy bookmarks, change the meaning of established commands like "print" and "view source", and in general make a mess of the user's prior understanding of the web.

- CSS is built on standard web models with cross-platform design and simple-to-understand codes that are precisely documented in public specifications. In contrast, frames are hard to learn, generally poorly documented, and have no chance of working on anything except desktop computers with relatively large screens (e.g., they are difficult to fit on personal display devices).

- Frames address the often-requested requirement to keep a part of the page from scrolling, though this one useful effect of frames can be achieved much simpler with the <BANNER> tag that was introduced in HTML 3.0. Most frame features are of relatively little value compared with the effects offered by tables and style sheets.

Problems with Frames

Highly skilled web designers can sometimes use frames to good effect, but even experienced designers are advised to use frames as sparingly as possible. The fundamental design of the web and web applications is based on having the page as the atomic unit of information, and the notion of the page permeates all aspects of the web. The page contains:

- The user's view of the information on the screen.
- Navigation
- A textual address used to retrieve information over the net (the URL)
- Storage of the information on the server and the author's editing unit (except if using embedded objects like image files that require the author to manage multiple files for a page).

Frames break this unified model of the web by introducing a way of looking at data that has not been well integrated into the other aspects of the web. With frames, the user's view of information

on the screen is determined by a sequence of navigation actions rather than a single navigation action. Navigation does not work with frames since the unit of navigation is different from the unit of view. If users create a bookmark in their browser they may not get the same view back when they follow the bookmark at a later date since the bookmark doesn't include a representation of the state of the frames on the page.

In addition to these fundamental problems, there are several minor problems with the implementation of frames.

Use of URLs

The addressing information shown at the top of the browser no longer constitutes a complete specification of the information shown in the window. If a page has a URL in a hypertext anchor in one of its pages then that anchor will not lead readers to the desired view but to the initial state of the frameset. Similarly, if a user decides to send an email message to a friend with the recommendation to check out a page, copying the URL from the browser will not work with frames since the URL points to the frameset and not to the current view (with the information of interest).

- To get URLs working, all hypertext links must have a TARGET="_top" attribute in their anchor tag (e.g., <A HREF=foo.html TARGET="_top">). Adding the _top forces the browser to clear out all the frames and replace the entire window with a new frameset. The destination frameset may have many frames that are identical to those in the departure frameset which will be cached in the browser, but by forcing a complete reload the browser gets a new URL for the destination. This means that navigation actions (e.g., bookmarks) work and the URL is available for reference by other people.

- The exception to using a TARGET="_top" attribute is when frames are used as a shortcut for scrolling within a single page. For example, a very long directory or other alphabetical listing could have a frame on top listing the letters of the alphabet. Clicking one of these letters would cause the listing to scroll within another frame while keeping the user on the same page and thus not destroying navigation.

Disabling Frames

For users with browsers that do not display frames or for which frames are turned off, it will not be possible to view the pages unless the page was programmatically altered to support no frames. Since most designers will not bother designing two versions of their pages and reserve <NOFRAMES> for a "helpful" link to the download site for a frames-supporting browser version, frames will not be useful for some target audience. The issues with Section 508 Compliance must also be considered when deciding the use of frames for web pages.

Navigating with the BACK button

This problem centers on the basic navigation of a web page. With some browsers, the BACK button simply does not work with frames. The BACK button is a safety net that gives users the confidence to navigate freely knowing that they can always get back to a safe point, where they started. Because the BACK button is one of the most used navigation features in web browsers, breaking the BACK button could be a major inconvenience for some users.

Print Problems

Many browsers cannot print framed pages appropriately. While most browsers cannot print anything really well, at least regular pages normally print in full. With frames, it is common to have the print command result in the printing of a single frame. Printing the full page is difficult with scrolling frames: should only the visible part of the frame be printed or should the content be allowed to expand and take up more room than it does on the screen?

Authoring Problems

Basic HTML is simple enough to learn so creating bug-free web pages is not a difficult task. Frames are another matter, though as they are currently so hard to learn that many page authors write buggy code.

Search Problems

Search engines have trouble with frames since they don't know what composites of frames to include as navigation units in their index.

User Preferences

Many websites that offer users a choice between regular and framed versions have found that most users prefer frame-free designs.

When to Use Frames

Frames are useful for "meta-pages" that comment on other pages. For example, a web design style guide may mix discussions of design principles with live examples of pages that follow (or break) the rules. In these cases, the embedded page should be treated as an embedded image (even though it is implemented as an independent page) and the "main" information that users will want to bookmark should be the content of the commenting frame. In general, do not use frames unless absolutely necessary and completely compliant. If they are used, then you must title each frame with text that facilitates frame identification and navigation.

Additional information related to this specific guideline may be found at:

http://www.access-board.gov/sec508/guide/1194.22.htm number(i)

http://www.w3.org/TR/WCAG10-TECHS/ numbergl-complex-elements

http://www.w3.org/WAI/wcag-curric/sam43-0.htm

D. 508 Compliance Testing

There are several tools available that can help you measure and identity your application's 508 compliance, however compliance must ultimately be determined by human interpretation and cannot be electronically automated. Below is a list of tools available for download free of charge. This list was provided by *http://www.usability.gov/accessibility/index.html numbertools.* At this time, the NCICB does not recommend any one tool.

Vischeck and Daltonize Color Blindness Tools
http://vischeck.com/index.php
- Vischeck simulates color blindness. Daltonize corrects images for colorblind users.
- Available as a free online tool, Photoshop plug-in, or Java application.

Delorie Lynx Viewer
http://www.delorie.com/web/lynxview.html
- Free web-based service simulating a text-only browser.
- Shows approximately how your pages will sound to a speech user.

Opera
http://www.opera.com/
- Free, fast, highly standards-compliant web browser.
- Allows you to toggle images, style sheets, scripting, and tables on and off with a single mouse click.
- Allows site managers to check alt-text, see how tables linearize, and ensure that pages work with style sheets turned off.

NIST WebMetrics Tool Suite
http://zing.ncsl.nist.gov/WebTools/
- A set of tools used to test the usability and accessibility of a site.

HTML Validation Service and CSS Validator
http://jigsaw.w3.org/css-validator/README.html
http://validator.w3.org/
- Minor HTML and CSS syntax errors can have serious consequences for accessibility.
- Free online validators from W3C help ensure your pages are error-free, accessible, and cross-platform compatible.

Access Adobe
http://access.adobe.com/
- Online tools for converting PDF files to HTML or ASCII.
- Resources for visually impaired users and more.

Bobby
http://bobby.watchfire.com/bobby/html/en/index.jsp
- Tool for checking web page accessibility and 508 compliance.

- Online service is free.
- Desktop version is paid.
- Requires an understanding of HTML.

W3C's List of Evaluation and Repair Tools

http://www.w3.org/WAI/ER/existingtools.html

- Descriptions and links to over 30 different kinds of tools.
- Help with evaluation, retrofitting, and transformation of web content.

E. Supported and Unsupported Technologies

The following table is a list of supported and unsupported technologies.

Section	Technology	Details	Supported
3.5 Frames 4.6 Use of Frames	Frames	*Avoid the use of frames.* If frames are justified by specific user needs, implement them with great caution and considerable design attention to ensure that they are laid out in a manner that ensures easy use by the visually impaired.	N
4.7 JavaScript	JavaScript	JavaScript cannot be used in place of business validation which must occur on the server side.	See Details
7.4 Image Maps	Image Maps	Server-side image maps are not supported.	See Details
7.11 Pop-Up Windows	Pop-up Windows	*Avoid pop-ups when possible.* If pop-ups are used, follow the guidelines in section 7.11.	N
13.2 Multimedia	Scalable Vector Graphics and Real Player	The use of SVG technology is supported as well as media that requires Real Player.	Y
13.2 Multimedia	Flash and Shockwave	Flash and Shockwave are discouraged.	N
13.3 Scripts	Scripts	When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology. *See example section 13.3.*	Y
13.4 Applets/Plug-ins	Applets/Plug-ins	Applets/Plug-ins are generally not relevant in our applications and *should be avoided.* When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).	N

F. Style Guides

The NCICB UI Group has recognized the ***Microsoft Manual of Style for Technical Publications*** as the recommended style guide for all NCICB developers. This guide is intended to give clear, up-to-date, and easy-to-use advice about usage and spelling of both general and computer-related terms, sentence style, technical writing issues, and design and interface elements.

Here are additional references and/or writings:

- The Chicago Manual of Style
- Elements of Style, William Strunk

G. NCICB Standards Checklist

The following checklist is intended to serve as a guide in determining NCICB Standards compliance. For additional information regarding individual requirements listed below, refer to the corresponding section.

Table 7 NCICB Standards Checklist

Section	Requirement	Mandatory - or - Optional	Met
3.1 Color	All information conveyed by color is also available without color, e.g., from context or markup.	M	
3.2 ALT Tags	All graphics must be coupled with an associated text equivalent.	M	
3.3 Skip Navigation	The application provides a method to facilitate the skipping of repetitive navigation.	M	
3.4 Flicker Rate	No screen flickers with a frequency greater than 2 Hz and lower than 55 Hz.	M	
3.5 Frames	Frames are avoided.	O	
3.7 Testing	Use automated accessibility compliance testing in addition to the use of individual user review.	M	
4.1 Internet Browsers	Design your application so it will render in Microsoft's Internet Explorer, Versions 5.0 and above, and Mozilla 1.0	M	
4.2 Color Schemes	All information conveyed by color is also available without color, e.g., from context or markup.	M	
4.3 Screen Resolution	All screens should be optimized for 1024X768 pixels resolution.	M	
4.4 Graphic Requirements	Avoid using pixilated, improperly compressed, or improperly scaled graphics.	M	
4.5 Style Sheets	Use of cascading style sheets throughout the application to simplify the consistent use of color and fonts.	O	
4.6 Use of Frames	Do not use frames, unless justified by specific user needs.	O	
4.7 Use of JavaScript	JavaScript is validated on the server side.	M	
4.8 Columns	All input controls (e.g., text boxes, drop-down lists, etc.) are identical in length when presented in a columnar format.	M	
4.9 Language Support	The basic language for all the NCICB web pages and applications is United States English.	M	

Section	Requirement	Mandatory - or - Optional	Met
5.1 Basic Layout			
5.1.1 Header Area	Every page contains the Cancer.gov mini- banner, as specified by cancer.gov web resources Design Standards, http://webresources.cancer.gov/designstandards.htm and must link to the Cancer.gov home page, http://cancer.gov.	M	
5.1.2 Navigation	"Help" is a default menu item.	M	
5.1.3 Page Content Area	The bulk of the display portion of a screen is comprised of data blocks that support the primary function of the screen. Fields usually derive from a single database record but can include multiple records or data from several tables.	M	
5.1.4 Action Area	Each action button shall be represented consistently with respect to size, shape, color, fonts, etc.	M	
5.1.4 Action Area	All buttons will have same height and unified width	M	
5.1.4 Action Area	All action buttons will contain a label	M	
5.1.4 Action Area	Action buttons that can cause the user to lose significant amounts of data, such as a 'Delete', should have a confirmation message associated with it	M	
5.1.4 Action Area	All the action buttons should use the standard button color	M	
5.1.5 Footer Area	The footer area should include contact us, Cancer.gov, NIH, HHS, FirstGov logos with hyperlinks, FAQ, contact us, Privacy Notice, Disclaimer, Accessibility, and Application Support.	M	
5.1.6 Disclaimers	The home page should include a disclaimer addressing privacy and retention of information collected.	M	
5.2 Default Information	If applicable, include log on/log off in the default area.	M	
5.3 Alternative Pages	All pages are accessible to visually impaired users. Pages that are not accessible must have a text only alternative.	M	
6.1 About this Application	About this Application is provided with version number.	M	
6.2 What's New	Provide information for the user community about new and exciting activities going on in and around this application and their community.	O	

Section	Requirement	Mandatory - or - Optional	Met
6.3 “Tip(s) of the Day”	“Tip(s) of the Day” will provide helpful tips that make use of the application simpler.	O	
6.4 Application Id	Identifier is easily found by the user it should be located in the bottom left-most corner of the content area.	M	
.1 Data Fields	Data fields should be associated with a text label.	M	
7.1 Data Fields	Fields containing numeric or decimal data should be right justified.	M	
7.1 Data Fields	When database fields are presented in a tabular or columnar format, each field should be clearly delineated from other fields in the row and column.	M	
7.1 Data Fields	Text fields should be left justified.	M	
7.1 Data Fields	Outline data fields with a plain box or bevel down box.	M	
7.1 Data Fields	Query Fields: used for searching, have a white background with black text.	M	
7.1 Data Fields	Editable Fields should be displayed with a white background and black font text, and should be marked with a ‘*’ for mandatory fields.	M	
7.1 Data Fields	When applicable, query fields should contain links for the first and last pages of data table.	M	
7.2 Data Blocks	Each block of data should be preceded with a horizontal title bar that separates this block from any previous blocks.	M	
7.2 Data Blocks	The block separator should be comprised of a block label, a "select all/none" control, and a "select all in a page" control.	M	
7.3 Field Labels	Field Labels should appear to the left of a database field when the screen is constructed in a single record format. When the screen presents data in a tabular or columnar format, text labels should appear at the top of the column, centered on the data. No colons should be used.	M	
7.4 Image Maps	Do not use Server-side image maps.	M	
7.4 Image Maps	If client-side image maps are used, provide redundant text links for each active region to support the visually impaired.	M	
7.5 Icons	Icons are large enough to be viewed on a screen with the standard screen resolution and are labeled intuitively.	M	
7.5 Icons	Icons are clearly and distinctly illustrate a function.	M	

Section	Requirement	Mandatory - or - Optional	Met
7.5 Icons	Icons that are repeated across pages should be positioned in the same place on each page on which the icon is displayed.	M	
7.6 Drop Down Lists	Drop-down lists are limited to no more than 15 items.	M	
7.6 Drop Down Lists	Drop-down lists are ordered according to some logical convention; in the absence of a specific convention, the list is alphabetized.	M	
7.7 Lists Boxes	A minimum of three choices are visible in all list boxes used.	M	
7.7 Lists Boxes	If multiple-selection is available for the list box, users are informed of this functionality.	M	
7.8 Radio Buttons and Check Boxes	Text labels are aligned left and the text is placed on the right side of the buttons.	M	
7.8 Radio Buttons and Check Boxes	Check boxes are exclusively to allow users to select one or more options from a known set of options.	M	
7.8 Radio Buttons and Check Boxes	Radio buttons and check boxes are aligned (vertically or horizontally) and put into groupings where appropriate.	M	
7.9 Push Buttons	Push Buttons must activate a specific function that cannot be incorporated into an Action Button.	M	
7.9 Push Buttons	Push Buttons do not navigate to a pop-up screen or related base screen. That should be handled through a navigation item.	M	
7.10 Scroll Bars	Data is controlled and displayed on a page-by-page basis rather than as a list that exceeds the length and width of the display area.	M	
7.11 Pop-Up Windows	Window Size does not exceed the base window dimensions.	M	
7.11 Pop-Up Windows	Pop-up windows are self-contained. The navigation function has to be completed within the scope of pop-up windows.	M	
7.11 Pop-Up Windows	A pop-up window should not be closeable or zoom able. To leave a pop-up window, the user has to press the "Close" or "Cancel" button.	M	
7.11 Pop-Up Windows	User cannot leave a pop-up window unless the database transaction is completed or canceled.	M	
8.1 Screen Navigation	When navigating single record blocks, the cursor will proceed to the first item of the next block when a Next Item command is issued and the cursor is in the last field of the block or the last item of the previous block when a Previous Item command is issued and the cursor is in the first field of the block.	M	

Section	Requirement	Mandatory - or - Optional	Met
8.1 Screen Navigation	Movement between pages occurs only as a result of a specific user action.	M	
8.1 Screen Navigation	The cursor does not skip display-only fields.	M	
8.1 Screen Navigation	The cursor can be positioned on buttons.	M	
8.1 Screen Navigation	When a new page is displayed, the cursor is positioned in the 1st editable field. If no editable fields exist, then the focus is given to the 1st selectable option.	M	
8.1 Screen Navigation	When the user presses the Tab key, the cursor moves to the next input field or action button.	M	
8.1 Screen Navigation	If possible, the Keyboard return is associate with the default action button (e.g., submit) on a page.	O	
9.1 Page Flows	If the number of records matching the criteria is more than 20, then the user is provided with navigation that allows them to navigate between the records using provided navigation.	M	
9.2 Queries and Hit Lists	Each hit list row is limited to a single displayed record. Multi-row displays are permitted only if a single display is not possible,	M	
9.2 Queries and Hit Lists	Hit Lists that contain more data rows than can appear on a single display screen are divided into discrete pages of information, rather then presenting the user with one long page of results.	M	
9.2 Queries and Hit Lists	If the data requires extensive retrieval time (e.g., 10+ seconds), the presentation is broken up to allow for a quicker display of the initial page.	M	
9.2 Queries and Hit Lists	In the event of a large data set, first page and last page buttons should be included in the navigation controls.	M	
9.2 Queries and Hit Lists	The shared hit list object supports customization.	M	
9.2 Queries and Hit Lists	Paging is consistent and predictable across the application.	M	
9.2 Queries and Hit Lists	The pages that offer a printer-friendly version of the content provide a button to open the printer-friendly version in a separate popup.	O	
9.2 Queries and Hit Lists	Nested sorts are used when users need to sort on more than one column in a hit list, or need to sort on more than one component of a column in a hit list.	M	

Section	Requirement	Mandatory - or - Optional	Met
9.2 Queries and Hit Lists	On a search screen, the ascending and descending radio buttons should be defaulted to ascending.	M	
9.2 Queries and Hit Lists	When a hit list is first displayed, the user or application defined default sort criteria will be displayed in the drop-down lists, including the ascending or descending order.	M	
9.2 Queries and Hit Lists	The developer chosen sort criteria will be displayed in the drop-down lists, including the ascending or descending order.	M	
9.2 Queries and Hit Lists	If a user selects the same sort in multiple drop-downs lists, a message indicating that the “sort reference is not valid” will be provided.	M	
10.1 Alert	Each page contains links to Page level and Systems level help in the menu bar. These links launch the independent help system into a separate browser.	M	
10.2 Context Sensitive Help	The application provides context-sensitive help (instructions, examples, etc.) for user input fields or groups of fields where the field is not self-explanatory.	M	
10.3 Context Sensitive FAQS	The user has access to a list of Frequently Asked Questions (FAQ) that are specific to the topic for which the user requested help.	M	
10.4 Page Level Help	A separate help page is available for each page of the application. Page level help should be the primary location for DETAILED information about specific tasks and functions on a particular page. Access to page level help will be through the help icon in the menu bar.	M	
10.5 System Level Help	A detailed table of contents for all of the help system is provided.	M	
10.5 System Level Help	Table of Contents for the FAQ system is provided.	M	
10.5 System Level Help	Help Index is provided.	M	
10.5 System Level Help	Glossary (if applicable) is provided.	M	
10.6 Hover Help	Descriptive information is provided whenever the mouse icon is place over or hovers over an image within the application.	O	
10.7 Descriptive Text	The application provides examples of the input being requested from the user wherever there is a possibility of choosing different formats.	M	

Section	Requirement	Mandatory - or - Optional	Met
10.8 Application Support	All pages contain a rendering DTG and screen identifier.	M	
10.8 Application Support	All pages contain the capability for the user to email the NCICB Application Support. When invoked, the Subject line of the email shall be pre-loaded with the user's current screen id.	M	
12.2 Error Messages	Error messages contain a (n):		
12.2 Error Messages	Explicit indication that something has gone wrong.	M	
12.2 Error Messages	Human-readable language, instead of obscure codes or abbreviations such as "a type 2 error has occurred."	M	
12.2 Error Messages	Precise descriptions of the exact problems, rather than vague generalities such as "syntax error."	M	
12.2 Error Messages	If relevant, include an error Id number (at the end of the error message) to assist NCICB Application Support personnel in determining and correcting the problem.	O	
12.3 Message Boxes	Avoid technical jargon, system-oriented information, error codes, etc. Avoid phrasing that blames the user or implies user error.	M	
13.1 Design Considerations	Avoid horizontal scrolling with a scroll bar.	M	
13.1 Design Considerations	Use vertical scroll bars and page up and page down to scroll up and scroll down sets of records at a time.	M	
13.1 Design Considerations	When leaving screens, transactions will be committed or discarded.	M	
13.1 Design Considerations	The menu changes dynamically with each major option but should not change for windows and pop-ups. Options are grayed out where not available.	M	
13.2 Multimedia Presentations	SVG technology is supported as well as media that requires Real Player; however Flash and Shockwave are discouraged.	M	
13.3 Scripts	Script information must be provided in a fashion that can be read by assistive technologies.	M	
13.4 Applets and Plug-ins	Applets and plug-ins should be avoided.	M	
13.5 Electronic Forms	Screen readers can relay the relevant field descriptors when tabbing from field to field.	M	
13.6 Performance			
13.6.1 Graphics	Do not use of graphics alone to convey information.	M	
13.6.2 Multimedia Objects	Multimedia effects are avoided unless it is relevant to the specific feature.	M	

13.6.3 Time Delays	The application shall not include any scripts that will cause a web page to disappear, “time out,” or expire if a response is not received within a specified time.	M	

If you have comments, questions, or change requests for this document, please contact NCICB Application Support.

NCICB Application Support
301-451-4384 or 888-478-4423
http://ncicb.nci.nih.gov/support
ncicb@pop.nci.nih.gov